Über Lösungen zu den WDVV-Gleichungen

- Diplomarbeit -

Konrad W. Schwerdtfeger

Institut für Theoretische Physik
Leibniz Universität Hannover

10. Mai 2010

Eingereicht am 10. Mai 2010

Betreuer und 1. Gutachter:	Prof. Dr. Olaf Lechtenfeld
2. Gutachter:	Prof. Dr. Norbert Dragon

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hannover, den 10. Mai 2010

Inhaltsverzeichnis

Einleitung

Diese Arbeit handelt von Lösungen der für die theoretische Physik bedeutenden WDVV-Gleichungen. Es wird ein kurzer Überblick über die Herkunft und die physikalische Bedeutung der Gleichungen gegeben, wichtige bisherige Erkenntnisse zu Lösungen eines speziellen Typs, den „Kovektor-Lösungen", werden zusammengestellt, und eine neue Methode zur Klassifikation und Konstruktion von Kovektor-Lösungen wird vorgestellt.

Die WDVV (Witten-Dijkgraaf-Verlinde-Verlinde)-Gleichungen sind ein System nichtlinearer partieller Differentialgleichungen für eine Funktion F von n reellen oder komplexen Variablen; Sie können geschrieben werden als[1]

$$F_i F_k^{-1} F_j = F_j F_k^{-1} F_i, \quad i,j,k = 1,...,n$$

mit den Matrizen

$$(F_i)_{lm} := \partial_i \partial_l \partial_m F.$$

Diese Gleichungen treten in verschiedenen Gebieten der theoretischen Physik auf, in 2-dimensionaler topologischer Feldtheorie, in Seiberg-Witten Theorie und in $\mathcal{N} = 4$ superkonformer Vielteilchenmechanik. Sie stehen somit in Zusammenhang mit einer Vielzahl von physikalischen Konzepten; in Kapitel 1 wird ein kleiner Streifzug gemacht, der einen Eindruck davon vermitteln soll.

Aufgrund ihrer großen Bedeutung in den genannten physikalischen Gebieten und aufgrund ihrer mathematischen Eigenschaften haben sich seit ihrer Entdeckung 1990 viele Physiker und Mathematiker mit ihnen beschäftigt, und Lösungen zu ihnen wurden gesucht. Eine Vielzahl von Lösungen teils sehr unterschiedlicher Typen wurde im Lauf der Jahre gefunden, sowohl aus den physikalischen Theorien als auch durch mathematische Analyse und Umformulierung. Bis heute ist es jedoch nicht gelungen, alle Lösungen zu klassifizieren und dies scheint auch überhaupt noch nicht absehbar. Man hat keine Vorstellung davon, wie viele weitere Lösungen existieren und was für andere Formen man finden könnte.

Diese Arbeit konzentriert sich auf Lösungen eines speziellen Typs, die *Kovektor-Lösungen;* sie haben die Form

$$F(x) = \sum_{\alpha \in \mathcal{A}} \alpha(x)^2 \log \alpha(x)^2$$

für $x \in V = \mathbb{C}^n$ mit einer endlichen Menge

$$\mathcal{A} \equiv \{\alpha\} \subset V^* \tag{0.1}$$

von Kovektoren. Da jede solche Funktion F durch „ihr System" $\mathcal{A}$ definiert ist, entspricht

[1] In dieser Form werden sie manchmal als verallgemeinerte WDVV-Gleichungen bezeichnet, da sie zuerst (in der topologischen Feldtheorie) nur zusammen mit einer weiteren Forderung an die Funktion F auftauchten und daher in einer „spezielleren" Form geschrieben wurden (siehe Kapitel 1). Außerdem werden weitere Darstellungen der allgemeinen Form verwendet.

die Suche nach WDVV-Kovektor-Lösungen der Suche nach den entsprechenden Systemen $\mathcal{A}$ von Kovektoren, $\vee$-Systeme genannt. Dies wird in Abschnitt 2.1.1 genauer erläutert.

Es ist gelungen, die von den WDVV-Gleichungen an diese Kovektor-Systeme gestellten Forderungen umzuformulieren in verhältnismäßig einfache geometrische Bedingungen, $\vee$-Bedingungen genannt. Verschiedene Darstellungen dieser Bedingungen werden in Abschnitt 2.1.2 zusammengestellt

Ferner reicht es, eine bestimmte Klasse von $\vee$-Systemen $\mathcal{A}$ zu finden, aus welcher sich alle anderen durch $GL(n)$-Transformationen ergeben. Für diese Klasse lauten die Bedingungen, wenn man V^* mit V identifiziert, wodurch die geometrische Bedeutung klarer wird

- Für jede Ebene in V, in welcher genau zwei Elemente α, β von $\mathcal{A}$ liegen, sind diese orthogonal, $\sphericalangle(\alpha, \beta) = \pi/2$
- Für jede Ebene Π in V, in welcher drei oder mehr Elemente von $\mathcal{A}$ liegen gilt, dass die Lineare Abbildung von $V \to V$, definiert durch

  $$G_\Pi := \sum_{\alpha \in \Pi \cap \mathcal{A}} \alpha \otimes \alpha$$

  in der Ebene Π proportional zur Identität ist,

  $$G_\Pi(x) = \lambda x \quad \forall x \in \Pi$$

  mit $\lambda \in \mathbb{R}$ bzw. $\lambda \in \mathbb{C}$, welches von der Ebene Π abhängen kann

Die Umformulierung der WDVV-Gleichungen für den Kovektor-Ansatz in diese geometrischen Bedingungen macht die systematische Suche nach Kovektor-Lösungen offensichtlich deutlich einfacher. Der Beweis der Äquivalenz ist nicht trivial und wurde von A. P. Veselov zwischen 1999 und 2007 in mehreren Schritten gefunden, er wird hier in Abschnitt 2.1.3 insgesamt dargestellt.

Kovektor-Lösungen wurden auf viele verschiedene Arten gefunden: Die ersten kamen aus der Seiberg-Witten-Theorie [15], dann fand man, dass alle Wurzelsystemen halbeinfacher Lie-Algebren $\vee$-Systeme darstellen [20], was mit den $\vee$-Bedingungen auf Coxeter-Wurzelsysteme verallgemeinert werden konnte [29]; weiterhin wurde festgestellt dass Deformationen von Wurzelsystemen, die aus der Theorie der Calogero-Moser bekannt waren, ebenfalls $\vee$-Systeme sind [29, 30, 31, 32]. Eine weitere Methode ist die Deformation von aus den Vektoren der Wurzelsystemen gebildeten Polytopen; die Wurzeln werden dabei als Kanten der Polytope aufgefasst [25, 26, 27, 38]. Viele neue $\vee$-Systeme hat man über die $\vee$-Bedingungen aus bekannten $\vee$-Systemen durch Projektion auf bestimmte Unterräume gefunden [33, 34]. Eine weitere Klasse liefern Deformationen „verallgemeinerter Wurzelsysteme" aus der Theorie der Lie-Superalgebren [34].

Es hat sich jedoch herausgestellt, dass alle bisher gefundenen $\vee$-Systeme aus den Coxeter-Wurzelsystemen, den verallgemeinerten Wurzelsystemen und Projektionen dieser auf Unterräume erhalten werden können. Die entsprechenden Definitionen und Beweise werden daher in Abschnitt 2.2. zusammengestellt.

Mit den genannten Methoden hat man eine Vielzahl von $\vee$-Systemen gefunden, sowohl einzelne Systeme als auch große Parameterfamilien, die teils sogar die Dimension des Vektorraums als Parameter besitzen. Die einfachsten Beispiele hierfür sind die Wurzelsysteme der klassischen Lie-Algebren, z.B. das A_n-System, das im $\mathbb{R}^{n+1}$ aus den Vektoren $e_i - e_j$, $i, j \in \{1, ..., n+1\}$ gebildete werden kann oder seine Deformationen

$$A_n(c) = \{\sqrt{c_i c_j}(e_i - e_j),\ 1 \leq i < j \leq n+1\},$$

parametrisiert durch $n \in \mathbb{N}$ und $c_1, ..., c_{n+1} \in \mathbb{R}_+$. Einzelne Systeme hat man z.B. aus den exzeptionelle Wurzelsystemen und ihren Projektionen erhalten, z.B. aus der Projektion von E_8 auf einen 6-dimensinalen Unterraum das System

$$e_i \pm e_j\,(1 \leq i < j \leq 6), \quad 2e_i\,(1 \leq i \leq 6), \quad \frac{1}{\sqrt{2}}(e_1 \pm e_2 \pm e_3 \pm e_4 \pm e_5 \pm e_6)$$

mit 91 Vektoren, die 1371 Ebenen aufspannen, wovon 940 zwei Vektoren enthalten, 416 drei und 15 vier.

Man hat es bisher aber nicht geschafft alle $\vee$-Systeme zu klassifizieren, die Situation ist hier die selbe wie beim Modulraum *aller* Lösungen zu den WDVV-Gleichungen, obwohl scheinbar ein viel einfacheres Problem vorliegt. Schon in Dimension 3 ist noch vollkommen unklar, was für weitere $\vee$-Systeme außer den bisher gefundenen möglich sind und womit sie sich klassifizieren lassen. Einen Überblick über die bisher bekannten $\vee$-Systeme in Dimension 3 und ihre Zusammenhänge über Parameter gibt die noch aktuelle Grafik am Ende von [34].

Das Ziel ist es natürlich, *alle* $\vee$-Systeme zu finden und zu klassifizieren. Dies ist mit den bisherigen Methoden jedoch nicht möglich, es scheint ein systematischerer Zugang nötig zu sein. Einen solchen stellt die Charakterisierung der $\vee$-Systeme, bzw. der Kandidaten für $\vee$-Systeme, über ihre Matroide [39, 40] oder Hypergraphen dar. Dies wurde 2009 von O. Lechtenfeld vorgeschlagen und in Zusammenarbeit mit Johannes Thürigen und dem Autor in Angriff genommen. Eine Einführung in die zugrunde liegenden Ideen un Begriffe gibt die Diplomarbeit von Johannes Thürigen [38].

Vom Autor wurde mit der Entwicklung und Implementierung eines Algorithmus zur systematischen Konstruktion von $\vee$-Systemen über Hypergraphen begonnen; unter anderem konnten damit bereits alle bekannten $\vee$-Systeme mit bis zu 10 Vektoren reproduziert werden und es konnte so gezeigt werden, dass alle $\vee$-Systeme mit bis zu 10 Vektoren schon gefunden wurden. Dieser Algorithmus wird in Kapitel 3 angegeben und kurz erläutert und ist außerdem auf der beiliegenden CD als funktionesfähige *Mathematica*-Datei enthalten.

1 Herkunft und physikalische Bedeutung der WDVV-Gleichungen

1.1 Topologische Feldtheorie

1.1.1 Herkunft der WDVV-Gleichungen

Die WDVV-Gleichungen traten erstmals in 2-dimensionaler topologischer Feldtheorie auf, als Bedingung an die generierende Funktion (das Präpotential) von Korrelationsfunktionen [1, 2]. Sie wurden aus Rekursionsrelationen für die n-Punkt Korrelationsfunktionen abgeleitet.

Wir beginnen hier wie E. Witten 1990 in [1] und betrachten die n-Punkt Korrelationsfunktionen

$$\left\langle \mathcal{O}_{\alpha_1}(x_1)\, \mathcal{O}_{\alpha_2}(x_2) \ldots \mathcal{O}_{\alpha_n}(x_n) \right\rangle_0 =: c_{\alpha_1\alpha_2\ldots\alpha_n} \tag{1.1}$$

für BRST-invariante lokale Operatoren (Null-Formen) $\mathcal{O}_\alpha$, $\alpha = 1, ..., N$, zunächst in Abwesenheit von Gravitation. Hier ist ein Operator als Identität ausgezeichnet, $\mathcal{O}_1 = \boldsymbol{1}$, und die Metrik ist durch die 2-Punkt Funktion

$$\left\langle \mathcal{O}_\alpha \mathcal{O}_\beta \right\rangle_0 = c_{1\alpha\beta} = \eta_{\alpha\beta}$$

auf der Sphäre gegeben. Weiter hat man die Operator-Produkt Algebra

$$\mathcal{O}_\alpha(x)\mathcal{O}_\beta(y) \to \sum_\gamma c_{\alpha\beta}{}^\gamma \mathcal{O}_\gamma(y), \qquad x \to y, \tag{1.2}$$

die in konform-invarianten Modellen dem Ring der lokalen Observablen entspricht (bekannt als der *chirale primäre Ring*).

Mittels der Operator-Produkt Algebra können die n-Punkt Funktionen (1.1) rekursiv auf 1-Punkt Funktionen reduziert werden: Da sie aufgrund topologischer Invarianz unabhängig von den x_i sind, kann man z.B. den Grenzwert $x_1 \to x_2$ nehmen und $\mathcal{O}_{\alpha_1}(x_1)\, \mathcal{O}_{\alpha_2}(x_2)$ durch $c_{\alpha\beta}{}^\gamma \mathcal{O}_\gamma(x_2)$ ersetzen[2] .

Insbesondere wird damit die 3-Punkt Funktion

$$\left\langle \mathcal{O}_\alpha \mathcal{O}_\beta \mathcal{O}_\gamma \right\rangle_0 = c_{\alpha\beta}{}^\sigma \eta_{\sigma\gamma} = c_{\alpha\beta\gamma}, \tag{1.3}$$

und für die 4-Punkt Funktion folgt durch unterschiedliche Anwendung der Operator-Produkt-Algebra (oder aus dem Faktorisierungsgesetz)

$$\left\langle \mathcal{O}_\alpha \mathcal{O}_\beta \mathcal{O}_\gamma \mathcal{O}_\delta \right\rangle_0 = c_{\alpha\beta}{}^\sigma c_{\sigma\gamma\delta} = c_{\alpha\gamma}{}^\sigma c_{\sigma\beta\delta}. \tag{1.4}$$

Die Gleichheit der beiden rechten beiden Ausdrücke bedeutet hier die Assoziativität der Multiplikation der lokalen Observablen,

$$(\mathcal{O}_\alpha \mathcal{O}_\beta)\mathcal{O}_\gamma = \mathcal{O}_\alpha(\mathcal{O}_\beta \mathcal{O}_\gamma). \tag{1.5}$$

[2] hier und in der gesamten Arbeit gelte die Summationskonvention

R. Dijkgraaf, H. Verlinde und E. Verlinde haben nun 1991 in [2] die Berücksichtigung der Gravitation mittels Störungstheorie betrachtet. Dazu sind allgemeinere Korrelationsfunktionen

$$\langle \mathcal{O}_{\alpha_1} ... \mathcal{O}_{\alpha_n} \int \mathcal{O}_{\alpha_1} ... \int \mathcal{O}_{\alpha_n} \rangle_0 \tag{1.6}$$

zu nehmen und man kann Kopplungskonstanten t_α für die Operatoren $\int \mathcal{O}_\alpha$ einführen, womit die gestörten 3-Punkt Funktionen die Form

$$c_{\alpha\beta\gamma}(t) = \langle \mathcal{O}_\alpha \mathcal{O}_\beta \mathcal{O}_\gamma \exp(\sum_\nu t_\nu \int \mathcal{O}_\nu) \rangle_0 \tag{1.7}$$

mit $t = (t_1, t_2, ..., t_n)$ annehmen. Obwohl dieses störungstheoretische Modell nicht mehr konform invariant ist, folgt auch hier aus der Konsistenz der Faktorisierungen der 4-Punkt Funktionen wieder, dass die 3-Punkt Funktionen $c_{\alpha\beta\gamma}(t)$ für alle Werte der t_α einen assoziativen Ring definieren, den *gestörten chiralen Ring*.

Ferner folgen zwei weitere Eigenschaften aus der Ward-Identität für das Spin-2 Feld $G(z)$:

- Die gestörte 2-Punkt Funktion ist unabhängig von den Kopplungskonstanten, so dass man eine konstante Metrik erhält,

  $$c_{1\sigma\tau}(t) = \eta_{\sigma\tau} \tag{1.8}$$

  mit der Inversen $(\eta_{\sigma\tau})^{-1} = \eta^{\sigma\tau}$; hierbei gehört der Index 1 zu dem Identitätsoperator $\mathcal{O}_1 \equiv \mathbf{1}$

- Die 3-Punkt Funktionen erfüllen Integrabilitätsbedingungen, sodass sie immer durch eine generierende Funktion F ausgedrückt werden können,

  $$c_{\alpha\beta\gamma}(t) = \partial_\alpha \partial_\beta \partial_\gamma F(t). \tag{1.9}$$

  Das Präpotential F stellt in topologischer Stringtheorie die freie Energie (in Tree-Näherung) dar.

In topologischen Modellen mit zentraler Ladung $d < 1$ folgt außerdem aus einer Auswahlregel, dass die $c_{\alpha\beta\gamma}(t)$ Polynome endlicher Ordnung in den t_α's sind, im allgemeinen ist lediglich von der Funktion F Quasihomogenität zu fordern,

$$F(\lambda^{d_1} t_1, ..., \lambda^{d_n} t_n) = \lambda^{d_F} F(t_1, ..., t_n) \tag{1.10}$$

für $\lambda \neq 0$ und $d_1, ..., .d_n$, $d_F \in \mathbb{R}$ [3].

Aus der Assoziativität des gestörten chiralen Rings folgt nun mit der konstanten Metrik (1.8), die Symmetrie der $c_{\alpha\beta\gamma}$ ausnutzend,

$$c_{\alpha\beta\sigma} \eta^{\sigma\tau} c_{\tau\gamma\delta} = c_{\alpha\delta\sigma} \eta^{\sigma\tau} c_{\tau\gamma\beta} \tag{1.11}$$

und somit schließlich aus (1.9) für das Präpotential F

$$\partial_\alpha\partial_\beta\partial_\sigma F\,\eta^{\sigma\tau}\,\partial_\tau\partial_\gamma\partial_\delta F = \partial_\alpha\partial_\delta\partial_\sigma F\,\eta^{\sigma\tau}\,\partial_\tau\partial_\gamma\partial_\beta F \quad \forall\alpha,\beta,\gamma,\delta \tag{1.12}$$

und

$$\partial_1\partial_\sigma\partial_\tau F = \eta_{\sigma\tau} = \text{const.} \tag{1.13}$$

(1.12) stellt für Funktionen, die (1.13) erfüllen die WDVV-Gleichungen dar.

1.1.2 WDVV-Lösungen aus der topologischen Feldtheorie

Um nun die Korrelationsfunktionen aus der topologischen Feldtheorie zu konstruieren und daraus WDVV-Lösungen zu erhalten, brauchen wir das gestörte Potential $W(x,t)$,

$$\mathcal{O}_\alpha(x,t) = -\frac{\partial}{\partial t_\alpha}W(x,t). \tag{1.14}$$

Der gestörte chirale Ring ist damit nun gegeben als der Polynomring der $\mathcal{O}_\alpha$'s modulo $W' \equiv \partial_x W$,

$$\mathcal{O}_\alpha(x,t)\mathcal{O}_\beta(x,t) = c_{\alpha\beta}{}^\gamma(t)\mathcal{O}_\gamma(x,t) \bmod W'(x,t), \tag{1.15}$$

wobei die Koeffizienten $c_{\alpha\beta}{}^\gamma$ mit den gestörten 3-Punkt Funktionen zu identifizieren sind via $c_{\alpha\beta\gamma} = \eta_{\gamma\delta}c_{\alpha\beta}{}^\delta$. Definiert man nun ein inneres Produkt auf dem Polynomring als

$$\langle\mathcal{O}_\alpha,\mathcal{O}_\beta\rangle = \oint\frac{dx}{2\pi\mathrm{i}}\frac{\mathcal{O}_\alpha(x)\mathcal{O}_\beta(x)}{W'(x)} \equiv \operatorname{res}\left(\frac{\mathcal{O}_\alpha\mathcal{O}_\beta}{W'}\right) \tag{1.16}$$

ergeben sich Residuenformeln für die 3-Punkt Funktionen,

$$c_{\alpha\beta\gamma} = \operatorname{res}\left(\frac{\mathcal{O}_\alpha\mathcal{O}_\beta\mathcal{O}_\gamma}{W'}\right). \tag{1.17}$$

Auf diese Weise hat man aus den Landau-Ginzburg Theorien Lösungen zu den WDVV-Gleichungen erhalten und auf ähnliche Weise von den Quanten-Kohomologien aus Deformationen des Polynomrings. Z.B. hat man in der Landau-Ginzburg Theorie ein einfaches Modell [18] mit $W'(x) = x^3 - q$, wofür sich aus den 3-Punkt Funktionen folgende Matrizen der dritten Ableitungen für F ergeben,

$$F_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & q \end{pmatrix}, \quad F_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & q \\ 0 & q & 0 \end{pmatrix}, \tag{1.18}$$

aus denen man durch Integration

$$F = \frac{1}{2}t_1t_2^2 + \frac{1}{2}t_1^2t_3 + \frac{q}{2}t_2t_3^2 \tag{1.19}$$

findet. (1.12) und (1.13) werden von F erfüllt, wie man mit (1.18) leicht überprüfen

kann. Für die Quanten-Kohomologie von $\mathrm{C}\mathbf{P}^2$ ergibt sich das Präpotential zu [18]

$$F = \frac{1}{2}t_1 t_2^2 + \frac{1}{2}t_1^2 t_3 + \sum_{k=1}^{\infty} \frac{N_k t_3^{3k-1}}{(3k-1)!} \mathrm{e}^{kt_2}, \tag{1.20}$$

wobei die Koeffizienten N_k, welche die Anzahl der rationalen Kurven angeben, hier gerade aus den WDVV-Gleichungen bestimmt werden können. Die entsprechenden Matrizen haben hier die Form

$$F_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & F_{222} & F_{223} \\ 0 & F_{223} & F_{233} \end{pmatrix}, \quad F_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & F_{223} & F_{233} \\ 0 & F_{233} & F_{333} \end{pmatrix}$$

und die WDVV-Gleichungen werden $F_{333} = F_{223}^2 - F_{222}F_{233}$, woraus sich folgende Rekursionsrelation für die N_k ergibt,

$$\frac{N_k}{(3k-4)!} = \sum_{a+b=k} \frac{a^2 b(3b-1)b(2a-b)}{(3a-1)!(3b-1)!} N_a N_b, \tag{1.21}$$

die 1994 von M. Kontsevich gefunden wurde [5].

Aufgrund ihrer Bedeutung für 2-dimensionale topologische Theorien und ihrer mathematischen Eigenschaften haben sich in den folgenden Jahren viele Physiker und Mathematiker mit den WDVV-Gleichungen beschäftigt, insbesondere B. Dubrovin [3], welcher eine bedeutsame differentialgeometrische Formulierung mittels Frobenius-Mannigfaltigkeiten entwickelt hat. Im Zusammenhang mit Hesseschen Metriken wurden sie z.B. von B. Totaro, H. Shima und K Yagi sowie H. Kito betrachtet [6, 7, 8].

Im Rahmen von Landau-Ginzburg Theorien wurden sie z.B. von I. Krichever untersucht, welcher auch die Bedeutung der Funktion F als Logarithmus der τ-Funktion von Whitham-Hierarchien geklärt hat [4], im Rahmen von Quanten-Kohomologien hat sich z.B. Yu. Manin mit ihnen beschäftigt [5]. Weiterhin wurden die Zusammenhänge mit Loop-Algebren, Kp-Hierarchien sowie Darboux-Egoroff Systemen von H. Aratyn, J. van de Leur und weiteren untersucht [9, 10, 11, 12, 13].

1.2 Seiberg-Witten Theorie

1.2.1 Allgemeine Form der WDVV-Gleichungen

1996 stellten G. Bonelli und M. Matone [14] und kurz danach A. Marshakov, A. Mironov und A. Morozov [15] fest, dass das Seiberg-Witten Präpotential von 4-dimensionalen $\mathcal{N}$=2 SUSY Yang-Mills Modellen die WDVV-Gleichungen erfüllt. Dieses Präpotential $\mathcal{F}$ unterliegt keiner „Normierungsbedingung" wie (1.13), sodass die WDVV-Gleichungen hier erstmals in ihrer allgemeinen Form

$$\mathcal{F}_i \mathcal{F}_k^{-1} \mathcal{F}_j = \mathcal{F}_j \mathcal{F}_k^{-1} \mathcal{F}_i \quad \forall i,j,k \tag{1.22}$$

mit den Matrizen

$$(\mathcal{F}_i)_{jk} = \partial_j \partial_j \partial_k \mathcal{F} \tag{1.23}$$

auftauchen.

Dass es hier keinen ausgezeichneten „ersten" Index gibt, spiegelt die Tatsache wider, dass es in Seiberg-Witten Theorie im Gegensatz zu den topologischen Feldtheorien aus dem vorigen Abschnitt keinen Einheitsoperator und keine ausgezeichnete Zeit-Variable t_1 gibt. Daher kann in der Seiberg-Witten-Theorie auch jede beliebige invertierbare Linearkombination der Matrizen $\mathcal{F}_i$ benutzt werden, um eine Metrik $G_{ij} = \sum g_{ij}^k \mathcal{F}_k$ bzw.

$$G = \sum g^k \mathcal{F}_k \tag{1.24}$$

zu definieren. Diese braucht nicht konstant zu sein, und die Koeffizienten g_{ij}^k können Funktionen der Koordinaten sein. Damit kann (1.22) schreiben als

$$\mathcal{F}_i G^{-1} \mathcal{F}_j = \mathcal{F}_j G^{-1} \mathcal{F}_i \quad \forall i, j, \tag{1.25}$$

was in Komponenten geschrieben zu

$$(\partial_i \partial_k \partial_p \mathcal{F}) G^{pq} (\partial_q \partial_l \partial_j \mathcal{F}) = (\partial_j \partial_k \partial_p \mathcal{F}) G^{pq} (\partial_q \partial_l \partial_i \mathcal{F}) \quad \forall i, k, j, l \tag{1.26}$$

wird (wobei die G^{ij} die Komponenten der inversen Metrik G^{-1} bezeichnet) , oder auch mit

$$C_i := G^{-1} \mathcal{F}_i \tag{1.27}$$

als Kommutatorgleichung

$$C_i C_j = C_j C_i \quad \forall i, j. \tag{1.28}$$

Beweis [20]: Die Äquivalenz von (1.25) und (1.28) für invertierbares G ist klar. Sei $\mathcal{F}$ Lösung von (1.22). Invertierung liefert $\mathcal{F}_j^{-1} \mathcal{F}_k \mathcal{F}_i^{-1} = \mathcal{F}_i^{-1} \mathcal{F}_k \mathcal{F}_j^{-1}$, das Bilden von Linearkombinationen ergibt $\mathcal{F}_j^{-1} G \mathcal{F}_i^{-1} = \mathcal{F}_i^{-1} G \mathcal{F}_j^{-1}$ und durch nochmaliges Invertieren erhält man (1.25). Sei nun umgekehrt $\mathcal{F}$ Lösung von (1.28). Dann folgt $G^{-1} \mathcal{F}_i \mathcal{F}_k^{-1} \mathcal{F}_j = C_i C_k^{-1} C_j = C_j C_k^{-1} C_i = G^{-1} \mathcal{F}_j \mathcal{F}_k^{-1} \mathcal{F}_i$, somit (1.22).

Man beachte, dass dies bedeutend weniger Gleichungen sind als (1.22). Für $G = \mathcal{F}_1$ erhält man insbesondere

$$\mathcal{F}_i \mathcal{F}_1^{-1} \mathcal{F}_j = \mathcal{F}_j \mathcal{F}_1^{-1} \mathcal{F}_i \quad \forall i, j, \tag{1.29}$$

was für die konstante Metrik (1.13)$\Longleftrightarrow$

$$(F_1)_{ij} = \eta_{ij} = \text{const} \tag{1.30}$$

gerade (1.12) entspricht. Hieraus erkennt man, dass die (verallgemeinerten) WDVV-Gleichungen (1.22)$\Longleftrightarrow$(1.25)$\Longleftrightarrow$(1.28)$\Longleftrightarrow$(1.29) allgemeinere Lösungen zulassen als die Kombination von (1.12) und (1.13) bzw. (1.29) und (1.30), welche von den Präpotentialen F der topologischen Theorien des vorigen Abschnitts erfüllt werden müssen.

1.2.2 Bedeutung der WDVV-Gleichungen in Seiberg-Witten Theorie

In der Seiberg-Witten Theorie kann der Niedrig-Energie Anteil der effektiven Wirkung von supersymmetrischen $\mathcal{N}$=2 Yang-Mills Modellen durch solch ein Präpotential $\mathcal{F}$ ausgedrückt werden [15, 16]. Das Präpotential wird dabei definiert durch eine Familie Riemannscher Flächen, ausgestattet mit meromorphen 1-Differentialen.

Als konkretes Beispiel betrachten wir die reine $SU(n)$ Yang-Mills Eichtheorie, welche durch die periodische Toda-Kette beschrieben werden kann [18]. Die Riemannsche Fläche $\mathcal{C}$ ist die spektrale Kurve des integrablen Systems, gegeben durch die Eigenwertgleichung

$$\det(L(w) - \lambda) = 0 \tag{1.31}$$

mit dem Lax-Operator

$$L(w) = \begin{pmatrix} p_1 & e^{q_1-q_2} & \cdots & w \\ e^{q_1-q_2} & p_2 & & \\ \vdots & & \ddots & \\ 1/w & & & p_n \end{pmatrix}. \tag{1.32}$$

Hieraus erhält man die Gleichungen

$$w + \frac{1}{w} = P(\lambda) = \prod_{i=1}^{n}(\lambda - \lambda_i), \quad \sum_i \lambda_i = 0. \tag{1.33}$$

Hierbei sind die λ_i Integrale der Bewegung und parametrisieren als Verzweigungspunkte den Modulraum der spektralen Kurven P. Ersetzt man in (1.33) $Y = w - 1/w$, erhält man die Standardform einer hyperelliptischen Riemannschen Fläche vom Geschlecht $n-1$,

$$Y^2 = P^2 - 4. \tag{1.34}$$

Das meromorphe 1-Differential wird damit

$$dS = \lambda \frac{dw}{w} = \lambda \frac{dP}{Y}, \tag{1.35}$$

und die Ableitungen von dS bzgl. der λ_i sind holomorphe 1-Differentiale auf der spektralen Kurve.

Das Präpotential wird implizit durch die Kohomologieklasse von dS definiert:

$$\begin{gathered} \frac{\partial \mathcal{F}}{\partial a_i} = a_i^D, \quad a_i = \oint_{A_i} dS, \quad a_j^D = \oint_{B_j} dS, \\ A_i \circ B_j = \delta_{ij} \end{gathered} \tag{1.36}$$

Die „B-Perioden" a_j^D sind hierbei dual zu den „A-Perioden" a_i. Die Ableitungen von dS

nach den a_i geben wieder holomorphe 1-Differentiale,

$$\frac{\partial dS}{\partial a_i} = d\omega_i. \tag{1.37}$$

Damit kann man nun für die dritten Ableitungen $\mathcal{F}_{ijk}$ eine Residuenformel analog zu (1.17) angeben:

$$\mathcal{F}_{ijk} = \operatorname*{res}_{d\omega=0} \frac{d\omega_i d\omega_j d\omega_k}{d\omega d\lambda} \tag{1.38}$$

Außerdem ist in den Koordinaten $\{a_i\}$ die Matrix der zweiten Ableitungen von $\mathcal{F}$

$$\frac{\partial^2 \mathcal{F}}{\partial a_i \partial a_j} = T_{ij} \tag{1.39}$$

gerade die Riemannsche Periodenmatrix der Kurve $\mathcal{C}$. Daher ist es möglich, das Präpotential mit dem Logarithmus der (trunkierten) τ-Funktion der integrablen Whitham-Hierarchie zu identifizieren, $\mathcal{F} = \log(\tau)$. Diese Analogie zu den Präpotentialen der 2-dimensionalen topologischen Theorien war ein erster Hinweis, dass auch das Seiberg-Witten Präpotential WDVV-Gleichungen erfüllen sollte [15].

Die WDVV-Gleichungen für $\mathcal{F}$ drücken nun die Assoziativität der von den holomorphen Differentialen $d\omega_i$ (1.37) gebildeten Algebren aus: Mit $C_{ij}{}^k = (C_i)_j{}^k$ und (1.38) ist (1.28) die Assoziativitäsbedingung

$$(d\omega_i d\omega_j) d\omega_k = d\omega_i (d\omega_j d\omega_k) \tag{1.40}$$

für die Algebra

$$d\omega_i d\omega_j = C_{ij}{}^k d\omega_k. \tag{1.41}$$

Nicht für alle Seiberg-Witten Präpotentiale sind diese Algebren assoziativ und daher erfüllen auch nicht alle die WDVV-Gleichungen; sofern die Riemannschen Flächen hyperelliptisch sind (wie im Fall der $SU(n)$-Eichgruppen) sind sie jedoch z.B. immer erfüllt. Weiterhin erfüllt der führende störungstheoretische Teil der Präpotentiale die WDVV-Gleichungen automatisch [18]. Für die 4-dimensionalen $SU(n)$ Modelle hat dieser z.B. die Form

$$\mathcal{F} = \frac{1}{4} \sum_{1 \le i < j \le n-1} (x_i - x_j)^2 \log(x_i - x_j)^2 + \frac{1}{4} \sum_{i=1}^{n-1} x_i^2 \log x_i^2 \tag{1.42}$$

und für die 5-dimensionalen die Form

$$\mathcal{F} = \frac{1}{4} \sum_{i,j} \left(\frac{1}{3}(x_i - x_j)^3 - \frac{1}{2} \mathrm{Li}_3(\mathrm{e}^{-2(x_i - x_j)}) \right) - \frac{n}{4} \sum_{i<j<k} x_i x_j x_k, \tag{1.43}$$

wobei Li_3 den Standard-Trilogarithmus bezeichnet. So erhält man eine Vielfalt von Lösungen zu den WDVV-Gleichungen aus der Seiberg-Witten Theorie. (1.42) ist das erste Beispiel für eine Lösung der Kovektor-Form und war 1999 für R. Martini und P.K.H. Gragert und A. P. Veselov der Ausgangspunkt für die Suche nach weiteren Lösungen

dieser Form.

Andersherum sind die WDVV-Gleichungen auch für das Verständnis der Seiberg-Witten Theorie von großer Bedeutung, z.B. decken sie Zusammenhänge mit den topologischen Feldtheorien auf. Daher wurden die WDVV-Gleichungen auch im Rahmen der Seiberg-Witten Theorie ausführlich betrachtet, außer von A. Marshakov, A. Mironov und A. Morozov von vielen weiteren. Z.B. hat J. M. Isidro mit Techniken aus der M-Theorie Seiberg-Witten Modelle mit nicht-hyperelliptischen Riemannschen Flächen konstruiert und gezeigt, dass auch diese eine assoziative Algebra besitzen und daher die WDVV-Gleichungen erfüllen könnten [19].

1.3 $\mathcal{N}$=4 superkonforme Vielteilchenmechanik

1.3.1 Konstruktion $\mathcal{N} = 4$ superkonformer Vielteilchenmodelle in $d = 1$

Die WDVV-Gleichungen liegen auch der Konstruktion von $\mathcal{N} = 4$ superkonformen quantenmechanischen Vielteilchenmodellen auf der reellen Geraden zugrunde, was S. Bellucci, A.V. Galajinsky und E. Latini 2004 festgestellt haben [21].

Dazu betrachten wir zunächst ein konformes quantenmechanisches System von $n + 1$ identischen Teilchen, die sich auf der reellen Geraden bewegen [25], mit Positionen x^I und Impulsen p_I ($I = 1, ..., n+1$). Der Hamiltonoperator hat (mit $m = 1$) die allgemeine Form

$$H = \frac{1}{2} p_I p_I + V_B(x^1, ..., x^{n+1}) \tag{1.44}$$

und ist für konform invariante Modelle Teil der $so(1, 2)$ konformen Algebra

$$[D, H] = -\mathrm{i}H, \quad [H, K] = 2\mathrm{i}D, \quad [D, K] = \mathrm{i}K \tag{1.45}$$

mit dem Dilatationsoperator D und dem konformen Boostgenerator K ($\hbar$=1). Ihre Darstellung in Form von Koordinaten und Impulsen bezüglich

$$[x^I, p_J] = \mathrm{i}\delta_J{}^I \tag{1.46}$$

lautet

$$D = -\frac{1}{4}(x^I p_I + p_I x^I), \quad K = \frac{1}{2} x^I x^I. \tag{1.47}$$

Die erste Relation in (1.45) entspricht der Forderung

$$(x^I \partial_I + 2) V_B = 0, \tag{1.48}$$

das Potential muss also für ein konform invariantes Modell homogen vom Grad -2 sein.

Fordert man nun zusätzlich Translations- und Permutationsinvarianz und lässt nur 2-Teilchen-Wechselwirkungen zu, erhält man das *Calogero-Modell* mit

$$V_B = \sum_{I<J} \frac{g^2}{(x^I - x^J)^2}. \tag{1.49}$$

Die $so(1,2) \simeq su(1,1)$ Algebra (1.45) soll nun zu der Superalgebra $su(1,1|2)$ mit einer zentralen Ladung erweitert werden. Der bosonische Teil von $su(1,1|2)$ enthält dann als Unteralgebra $so(1,2)$ und zusätzlich die $su(2)$ R-Symmetrie-Algebra, welche durch J_a, $a = 1,2,3$ erzeugt wird. Der fermionische Teil besteht aus den $su(2)$ Supersymmetrie-Generatoren Q_α und $\bar{Q}^\alpha$ zusammen mit ihren superkonformen Partnern S_α und $\bar{S}^\alpha$, $\alpha = 1,2$. Diese erfüllen die Relationen

$$(Q_\alpha)^\dagger = \bar{Q}^\alpha, \quad (S_\alpha)^\dagger = \bar{S}^\alpha, \tag{1.50}$$

die bosonischen Generatoren sind hermitesch. Die nicht-verschwindenden (anti-)Kommutatoren der superkonformen Algebra sind

$$\begin{aligned}
&[D,H] = -\mathrm{i}H && [H,K] = 2\mathrm{i}D \\
&[D,K] = \mathrm{i}K && [J_a,J_b] = \mathrm{i}\epsilon_{abc}J_c \\
&\{Q_a,\bar{Q}^\beta\} = 2H\delta_\alpha{}^\beta && \{Q_\alpha,\bar{S}^\beta\} = 2\mathrm{i}(\sigma_a)_\alpha{}^\beta J_a - 2D\delta_\alpha{}^\beta - \mathrm{i}C\delta_\alpha{}^\beta \\
&\{S_a,\bar{S}^\beta\} = 2K\delta_\alpha{}^\beta && \{\bar{Q}^\alpha,S_\beta\} = -2\mathrm{i}(\sigma_a)_\beta{}^\alpha J_a - 2D\delta_\beta{}^\alpha - \mathrm{i}C\delta_\beta{}^\alpha \\
&[D,Q_\alpha] = -\frac{1}{2}\mathrm{i}Q_\alpha && [D,S_\alpha] = \frac{1}{2}\mathrm{i}S_\alpha \\
&[K,Q_\alpha] = \mathrm{i}S_\alpha && [H,S_\alpha] = -\mathrm{i}Q_\alpha \\
&[J_a,Q_\alpha] = -\frac{1}{2}(\sigma_a)_\alpha{}^\beta Q_\beta && [J_a,S_\alpha] = -\frac{1}{2}(\sigma_a)_\alpha{}^\beta S_\beta \\
&[D,\bar{Q}^\alpha] = -\frac{1}{2}\mathrm{i}\bar{Q}^\alpha && [D,\bar{S}^\alpha] = \frac{1}{2}\mathrm{i}\bar{S}^\alpha \\
&[K,\bar{Q}^\alpha] = \mathrm{i}\bar{S}^\alpha && [H,\bar{S}^\alpha] = -\mathrm{i}\bar{Q}^\alpha \\
&[J_a,\bar{Q}^\alpha] = \frac{1}{2}\bar{Q}^\beta(\sigma_a)_\beta{}^\alpha && [J_a,\bar{S}^\alpha] = \frac{1}{2}\bar{S}^\beta(\sigma_a)_\beta{}^\alpha
\end{aligned} \tag{1.51}$$

mit $\epsilon_{123} = 1$, den Pauli-Matrizen $\sigma_1, \sigma_2, \sigma_3$ und der zentralen Ladung C.

Für eine mechanische Realisierung von $su(1,1|2)$ müssen nun die bosonischen Koordinaten x^I mit fermionischen Freiheitsgraden gepaart werden, dargestellt durch die zueinander hermitesch konjugierten Operatoren ψ^I_α und $\bar{\psi}^{I\alpha}$ mit $I = 1,...,n+1$ und $\alpha = 1,2$, mit den Antikommutatoren

$$\{\psi^I_\alpha, \psi^J_\beta\} = 0, \quad \{\bar{\psi}^{I\alpha}, \bar{\psi}^{J\beta}\} = 0, \quad \{\psi^I_\alpha, \bar{\psi}^{J\beta}\} = \delta_\alpha{}^\beta \delta^{IJ}. \tag{1.52}$$

Die Konstruktion von freien fermionischen Generatoren, bzgl. des freien Hamiltonoperators

$$H_0 = \frac{1}{2}p_I p_I, \tag{1.53}$$

ist in diesem erweiterten Raum einfach,

$$Q_{0\alpha} = p_I \psi^I_\alpha, \quad \bar{Q}^\alpha_0 = p_I \bar{\psi}^{I\alpha}, \quad S_{0\alpha} = x_I \psi^I_\alpha, \quad \bar{S}^\alpha_0 = x_I \bar{\psi}^{I\alpha}, \tag{1.54}$$

die entsprechenden $su(2)$ Generatoren werden

$$J_{0a} = \frac{1}{2}\bar{\psi}^{I\alpha}(\sigma_a)_\alpha{}^\beta\psi^I_\beta \tag{1.55}$$

und sind automatisch Weyl-geordnet. Die freien Dilatations- und konformen Boostgeneratoren behalten ihre bosonische Form (1.47). Im Gegensatz zum Fall von $\mathcal{N} \leq 2$ erfüllen die freien Generatoren jedoch nicht die volle Algebra (1.51). Sogar für $C = 0$ erfordern die Gleichungen für $\{Q, \bar{S}\}$ und $\{\bar{Q}, S\}$ Korrekturen der fermionischen Generatoren. Diese können auf Q und $\bar{Q}$ beschränkt werden,

$$Q_\alpha = Q_{0\alpha} - \mathrm{i}[S_{0\alpha}, V], \quad \bar{Q}^\alpha = \bar{Q}^\alpha_0 - \mathrm{i}[\bar{S}^\alpha_0, V] \tag{1.56}$$

mit $V \neq 0$,

$$H = H_0 + V. \tag{1.57}$$

Das heißt also, dass keine freie mechanische Realisierung von $su(1,1|2)$ möglich ist. Weiterhin ergibt sich, dass V in den Fermionen quadratische und quartische Terme enthalten muss,

$$V = V_B - U_{IJ}(x)\langle\psi^I_\alpha\bar{\psi}^{J\alpha}\rangle + \frac{1}{4}F_{IJKL}(x)\langle\psi^I_\alpha\psi^{J\alpha}\bar{\psi}^{K\beta}\bar{\psi}^L_\beta\rangle \tag{1.58}$$

mit total symmetrischen, vom Grad -2 in $x \equiv (x^1, ..., x^{n+1})$ homogenen, unbekannten Funktionen U_{IJ}, F_{IJKL} und V_B. Das Symbol $\langle ... \rangle$ steht hier für symmetrische (Weyl-) Ordnung. Die Unbestimmtheit der Ordnung im fermionischen Sektor wirkt sich auf das bosonische Potential V_B aus. Damit werden die Supersymmetrie-Generatoren

$$\begin{aligned} Q_\alpha &= (p_J - \mathrm{i}x^I U_{IJ}(x))\psi^J_\alpha - \frac{\mathrm{i}}{2}x^I F_{IJKL}(x)\langle\psi^J_\beta\psi^{K\beta}\bar{\psi}^L_\alpha\rangle, \\ \bar{Q}_\alpha &= (p_J + \mathrm{i}x^I U_{IJ}(x))\psi^J_\alpha - \frac{\mathrm{i}}{2}x^I F_{IJKL}(x)\langle\psi^{J\alpha}\psi^{K\beta}\bar{\psi}^L_\beta\rangle. \end{aligned} \tag{1.59}$$

Einsetzen von (1.57), (1.59) und $D = D_0$, $K = K_0$, $J_a = J_{0a}$, $S_\alpha = S_{0\alpha}$, $\bar{S}_\alpha = \bar{S}^\alpha_0$ in die Algebra (1.51) liefert nun eine lange Liste von Bedingungen an V_B, U_{IJ} und F_{IJKL}. Eine Konsequenz ist, dass

$$U_{IJ} = \partial_I\partial_J U, \quad F_{IJKL} = \partial_I\partial_J\partial_K\partial_L F, \tag{1.60}$$

womit zwei skalare Präpotentiale eingeführt werden. Die restlichen Bedingungen werden dann zu den so genannten Strukturgleichungen

$$(\partial_I\partial_K\partial_P F)(\partial_J\partial_L\partial_P F) = (\partial_J\partial_K\partial_P F)(\partial_I\partial_L\partial_P F), \qquad x^I\partial_I\partial_J\partial_K F = -\delta_{JK}, \tag{1.61}$$

$$\partial_I\partial_J U - (\partial_I\partial_J\partial_K F)\partial_K U = 0, \qquad x^I\partial_I U = -C. \tag{1.62}$$

1.3.2 Die WDVV-Gleichungen als Teil der Strukturgleichungen

Der linke Teil von (1.61) stellt die WDVV-Gleichungen dar für Funktionen F, für welche es eine Metrik (1.24) gibt, die proportional zur Einheitsmatrix ist, wie man durch Vergleich mit den allgemeinen WDVV-Gleichungen in der Form (1.26) erkennt. Dies ist hier aber aufgrund der rechten Gleichung in (1.61) immer der Fall:

$$G := x^I F_I. \tag{1.63}$$

Daher kann der linke Teil von (1.61) hier mit den WDVV-Gleichungen identifiziert werden. Sie können verstanden werden als Forderung nach verschwindender Krümmung für einen Zusammenhang $\partial^3 F$ auf dem Konfigurationsraum. Die linke Bedingung von (1.62) entspricht dann der Forderung nach kovarianter Konstanz von ∂U bzgl. $\partial^3 F$ und ihre Integrabilität impliziert die WDVV-Gleichungen projiziert auf ∂U. Der rechten Gleichungen in (1.61) und (1.62) stellen Homogenitätsbedingungen für F und U dar, sie sind inhomogen und explizit von den Koordinaten abhängig. Für Lösungen der Strukturgleichungen hat man offenbar immer die Freiheit der Addition eines quadratischen Polynoms zu F und einer Konstanten zu U.

Die rechte Gleichung in (1.61) kann nun zweimal integriert werden, wodurch man unter Nullsetzten der Integrationskonstanten - einer linearen Funktion auf der rechten Seite -

$$x^I \partial_I F - 2F + \frac{1}{2} x^I x^I = 0 \tag{1.64}$$

erhält. In dieser Form erkennt man sofort, dass die triviale WDVV-Lösung für F - ein quadratisches Polynom - hier ausgeschlossen ist.

Zur Vereinfachung der Strukturgleichungen ist es sinnvoll, die Schwerpunkts- und Relativbewegung der Teilchen zu separieren, was mit folgender Transformation möglich ist,

$$\{x^I\} \to \{x^i, X\},\ i = 1, ..., n \quad \text{mit} \quad X = \frac{1}{\sqrt{n+1}} \sum_{i=1}^{n+1} x^I, \tag{1.65}$$

womit die Relativkoordinaten x^i in der Hyperebene senkrecht zur Bewegung des Schwerpunkts eingeführt werden (ab hier betrachten wir nur noch die Relativbewegung, sodass es zu keiner Verwirrung über die Koordinaten x kommen kann). Die Strukturgleichungen gelten dann für beide Bewegungen unabhängig, und die Präpotentiale und die zentrale Ladung spalten sich entsprechend auf ,

$$F = F_{\text{com}}(X) + F_{\text{rel}}(x), \quad U = U_{\text{com}}(X) + U_{\text{rel}}(x) \quad \text{und} \quad C = C_{\text{com}} + C_{\text{rel}}, \tag{1.66}$$

wo nun $x \equiv \{x^i\}$. Für die Schwerpunktsbewegung ist die Lösung trivial,

$$F_{\text{com}} = -\frac{1}{2} X^2 \log|X|, \quad U_{\text{com}} = -C_{\text{com}} \log|X|. \tag{1.67}$$

Für die Relativkoordinaten ist in den Strukturgleichungen einfach $I, J, ...$ durch $i, j, ...$ und C durch C_{rel}zu ersetzten.

Für gegebene Präpotentiale F und U ist das superkonforme Modell dann vollständig bestimmt über (1.58) zusammen mit[3]

$$V_B = \frac{1}{2}(\partial_i U)(\partial_i U) + \frac{\hbar^2}{8}(\partial_i\partial_j\partial_k F)(\partial_i\partial_j\partial_k F), \tag{1.68}$$

$$x^i F_{ijkl} = -\partial_j\partial_k\partial_l F, \quad x^i U_{ij} = -\partial_i U. \tag{1.69}$$

Um nun aus den Strukturgleichungen superkonforme Modelle zu erhalten, gibt es zwei Möglichkeiten: Zum einen kann man von einem bekannten konformen Modell ausgehen, z.B. dem Calogero-Modell (1.49), aus seinem bosonischen Potential V_{B0} über

$$\frac{1}{2}(\partial_i U)(\partial_i U) = V_{B0} \tag{1.70}$$

und der Homogenitätsbedingung in (1.62), $x^i\partial_i U = -C$, ein Präpotential U bestimmen und dann dazu passende F suchen. Alternativ kann man von Lösungen F zu (1.61) ausgehen und dann versuchen, aus (1.62) ein dazu passendes U zu bestimmen.

Wie man aus (1.68) erkennt, liefert auch $U \equiv 0$, entsprechend $V_{B0} \equiv 0$ und einer verschwindenden zentralen Ladung C, nicht-triviale superkonforme Quantenmodelle. Diese können somit direkt aus WDVV-Lösungen gewonnen werden, welche die Homogenitätsbedingung in (1.61), äquivalent zu (1.64) bzw.

$$(x^i\partial_i - 2)F = -\frac{1}{2}x^i x^i, \tag{1.71}$$

erfüllen. Weiterhin folgt aus der Invarianz der WDVV-Gleichungen unter $GL(n)$-Transformationen, dass jede WDVV-Lösung mit *konstanter* Metrik (1.63) in eine Lösung mit $G \equiv \mathbf{1}$ transformiert werden kann. Insofern kann man natürlich auch gleich alle durch $GL(n)$-Transformationen zusammenhängenden Lösungen mit konstantem G identifizieren, womit dann jeder solchen Äquivalenzklasse genau ein superkonformes Modell mit $U \equiv 0$ entspricht.

Man hat hier eine ähnliche Situation wie in der 2-dimensionalen topologischen Feldtheorie, dass man die WDVV-Gleichungen in der Form (1.25) mit einem konstanten G schreiben kann, jedoch gibt es hier keine ausgezeichente „1." Koordinate.

Solche WDVV-Lösungen mit konstantem G liefert z.B. der Kovektor-Ansatz, der im 2. und 3. Kapitel behandelt wird. Die allgemeine Lösung zur Homogenitätsbedingung (1.71) kann (unter Weglassen des unbedeutenden Polynoms vom Grad 2) geschrieben werden als

$$F(x) = -\frac{1}{4}\sum_s f_s Q_s(x)\log|Q_s(x)| \tag{1.72}$$

mit reellen Koeffizienten f_s und quadratischen Formen

$$Q_s(x) = \sum_{i,j} q_{ij}^{(s)} x^i x^j \quad \text{mit} \quad \sum_s f_s Q_s(x) = x^i x^i. \tag{1.73}$$

[3]Hier wurde $\hbar$ wiederhergestellt, um zu zeigen, dass der Beitrag von F klassisch verschwindet

WDVV-Lösungen F, welche von dieser Form sind repräsentieren also Äquivalenzklassen mit konstantem G und liefern superkonforme Modelle.

Kovektor-Lösungen sind aus Termen mit quadratischen Formen vom Rang 1,

$$Q_\alpha(x) = (\alpha \cdot x)^2 = \sum_{i,j} \alpha_i \alpha_j x^i x^j, \quad \alpha = (\alpha_1, ..., \alpha_n), \tag{1.74}$$

zusammengesetzt. Für solche Lösungen

$$F_{\{\alpha\}} = -\frac{1}{4} \sum_\alpha f_\alpha Q_\alpha \log |Q_\alpha| \tag{1.75}$$

kann die Menge $\{\alpha\}$ der Kovektoren als irreduzibel, d.h. nicht in gegenseitig orthogonale Kovektormengen zerlegbar, vorausgesetzt werden, da

$$F_{\{\alpha\}\dot\oplus\{\beta\}} = F_{\{\alpha\}} + F_{\{\beta\}}. \tag{1.76}$$

Für ein reduzibles Kovektor-System

$$\{\alpha\} = L = L_1 \dot\oplus L_2 \dot\oplus \cdots \dot\oplus L_M \tag{1.77}$$

kann man jedoch durch „Kopplung" der F_{L_i} der Form (1.75) mit „relativen radialen Termen", bei denen Q von der Form

$$r_I^2 = \sum_{i \in S_I} x_i^2, \quad r_{IJ}^2 = \sum_{i \in S_I \cup S_J} x_i^2, \quad \ldots, \quad R^2 = \sum_{i \in \{1,2,...,n\}} x_i^2, \quad I, J, ... = 1, ..., M \tag{1.78}$$

ist mit der Zerlegung

$$\{1, 2, ..., n\} = \{1, ..., n_1\} \cup \{n_1+1, ..., n_1+n_2\} \cup \cdots \cup \{n-n_M+1, ..., n\} = S_1 \cup S_2 \cup \cdots \cup S_M, \tag{1.79}$$

entsprechend der Zerlegung (1.77) des Kovektor-Systems, neue, nicht reduzible, Lösungen

$$F = -\frac{1}{4} \sum_{\alpha \in L} f_\alpha (\alpha \cdot x)^2 \log |\alpha \cdot x| - \frac{1}{2} \sum_{I=1}^{M} f_{r_I} r_I^2 \log r_I^2 - \frac{1}{2} \sum_{I<J}^{M} f_{r_{IJ}} r_{IJ}^2 \log r_{IJ}^2 - \ldots - \frac{1}{2} f_R R^2 \log R^2 \tag{1.80}$$

erhalten. Diese Methode, neue Lösungen aus Kovektor-Lösungen zusammenzusetzten wurde 2009 von O. Lechtenfeld und K. Polovnikov gefunden [28].

Das Interesse an $\mathcal{N} = 4$ superkonformen Vielteilchenmodellen in einer Dimension kommt zum Teil von der noch unvollständig verstandenen AdS/CFT Korrespondenz. Eine interessante damit zusammenhängende Anwendung könnte in der Theorie Schwarzer Löcher liegen, da mit Hilfe des Anti-de-Sitter Raums die Geometrie einer großen Klasse extremer schwarzer Löcher beschrieben werden kann (siehe z.B. die Referenzen in [24]). Besonders interessant scheint die Vermutung aus [23], dass eine $\mathcal{N} = 4$ superkonforme Erweiterung des Calogero-Modells eine mikroskopische Beschreibung der

Geometrie extremer Schwarzer Löcher vom Reissner-Nordström Typ nahe des Horizonts liefern könnte, da deren Isometriegruppe gerade $su(1,1|2)$ ist. Das Calogero-Modell ist jedoch nicht das einzige exakt lösbare konforme Vielteilchenmodell in $d = 1$, und da es im Kontext von [23] nur auf die Struktur der konformen Algebra ankommt, scheint jede superkonforme $\mathcal{N} = 4$ Vielteilchenmechanik ein guter Ausgangspunkt.

2 WDVV-Lösungen der Kovektor-Form und Veselovsche ∨-Systeme

2.1 WDVV-Gleichungen und ∨-Bedingungen

2.1.1 Der Kovektor-Ansatz

Die WDVV-Gleichungen wurden im Rahmen des Kovektor-Ansatzes sowohl im Reellen als auch im Komplexen betrachtet. Im Komplexen ist der Kovektor-Ansatz

$$F(x) = \sum_{\alpha \in \mathcal{A}} \alpha(x)^2 \log \alpha(x)^2 \tag{2.1}$$

für $x \in V = \mathbb{C}^n$ mit einer endlichen Menge

$$\mathcal{A} \equiv \{\alpha\} \subset V^* \tag{2.2}$$

von p paarweise linear unabhängigen Kovektoren, welche V^* aufspannt. Jede Funktion F der Form (2.1) ist definiert durch ihr *System* $\mathcal{A}$ von Kovektoren. Löst F die WDVV-Gleichungen, so heiße $\mathcal{A}$ ∨-*System*, die Suche nach Kovektor-Lösungen entspricht also der Suche nach ∨-Systemen.

Im Reellen kann man entweder den Ansatz (2.1), (2.2) mit $V = \mathbb{R}$ betrachten oder statt (2.1) allgemeiner

$$F(x) = \sum_{\alpha \in \mathcal{A}} f_\alpha \alpha(x)^2 \log \alpha(x)^2 \tag{2.3}$$

mit *Multiplizitäten* $f_\alpha \in \mathbb{R}$ nehmen, wodurch Summanden mit negativem Vorzeichen möglich sind. Im Komplexen sind die Ansätze (2.1) und (2.3) äquivalent: Mit $\alpha \to \sqrt{f_\alpha}\alpha$ wird $F(x)$ in (2.1) zu

$$\sum_{\alpha \in \mathcal{A}} f_\alpha \alpha(x)^2 \log \alpha(x)^2 + \sum_{\alpha \in \mathcal{A}} (f_\alpha \log f_\alpha) \alpha(x)^2, \tag{2.4}$$

wovon der zweite Teil ein quadratisches Polynom in x und somit irrelevant ist. Im Zusammenhang mit dem Ansatz (2.3) soll hier das Paar $(\mathcal{A}, \{f_\alpha\})$ ∨-System genannt werden.

2.1.2 Formulierungen der ∨-Bedingungen

Ein entscheidender Schritt zum Verständnis und Auffinden von Kovektor-Lösungen ist die Umformulierung der allgemeinen WDVV-Gleichungen in die ∨-*Bedingungen* genannten geometrischen Forderungen an das System $\mathcal{A}$ von Kovektoren, wie wir in Abschnitt 2.2 und in Kapitel 3 sehen werden. Diese erstaunlich einfachen Bedingungen wurden 1999 von A.P. Veselov in [29] gefunden, im Zusammenhang mit der Feststellung, dass sowohl Coxeter-Wurzelsysteme als auch bestimmte, in der Theorie der Calogero-Moser Systeme entdeckte, Deformationen dieser Wurzelsysteme WDVV-Kovektor-Lösungen liefern. Es sind verschiedene Formulierungen der ∨-Bedingungen nützlich, hier werden einige dargestellt.

Sei $V = \mathbb{C}^n$ und ein System $\mathcal{A}$ von Kovektoren wie in (2.2) sei gegeben. Wir betrachten nun die Schnittmengen Π von $\mathcal{A}$ mit (2-dimensionalen) Ebenen $\pi \in V^*$, welche mehr als 2 Elemente besitzen. Anders gesagt sind dies die maximalen Untermengen von $\mathcal{A}$, die 2-dimensionale Unterräume aufspannen. In der Matroid-Theorie werden sie *2-flats* genannt. Sie bilden ein *System* in der Potenzmenge von $\mathcal{A}$,

$$\{\Pi\} \subset \mathcal{P}(\mathcal{A}). \tag{2.5}$$

Weiter seien

$$G := \sum_{\alpha\in\mathcal{A}} \alpha\otimes\alpha, \quad G_\Pi := \sum_{\alpha\in\Pi} \alpha\otimes\alpha, \tag{2.6}$$

auffassbar als 2-Formen $V \times V \to \mathbb{C}$ oder als lineare Abbildungen $V \to V^*$ bzw. als Matrizen mit Komponenten

$$G_{ij} = \sum_{\alpha\in\mathcal{A}} \alpha_i\alpha_j, \quad (G_\Pi)_{ij} := \sum_{\alpha\in\Pi} \alpha_i\alpha_j. \tag{2.7}$$

G ist invertierbar in V, die G_Π's jeweils in der von Π aufgespannten Ebene π. Für $\alpha \in V^*$ sei

$$\alpha^\vee := G^{-1}(\alpha) = G^{-1}\alpha \in V. \tag{2.8}$$

Damit ist z.B.

$$\alpha(\beta^\vee) = \beta(\alpha^\vee) = G(\alpha^\vee, \beta^\vee) = \alpha G^{-1}\beta. \tag{2.9}$$

Wir sagen, dass $\mathcal{A}$ die $\vee$-*Bedingungen* erfüllt, wenn eine der folgenden äquivalenten Gleichungen für jedes Paar (α, Π) mit $\alpha \in \Pi$ gilt,

$$G_\Pi\alpha^\vee = \lambda G\alpha^\vee \tag{2.10}$$

$$G_\Pi G^{-1}\alpha = \lambda\alpha \tag{2.11}$$

$$\sum_{\beta\in\Pi} \beta(\alpha^\vee)\beta = \lambda\alpha \tag{2.12}$$

mit $\lambda \in \mathbb{C}$, welches von Π und von α abhängen kann. Entsprechend (2.11) kann man also sagen, dass für jedes 2-flat Π die Vektoren α mit $\alpha^\vee \in \Pi$ die Eigenvektoren von $G_\Pi G^{-1}$ mit Eigenwerten λ sein müssen.

Hat ein 2-flat nur 2 Elemente, α, β, so muss entsprechend (2.12)

$$\beta(\alpha^\vee) = G(\alpha^\vee, \beta^\vee) = 0 \quad \forall\, \Pi = \{\alpha, \beta\} \tag{2.13}$$

gelten, d.h. $\alpha^\vee$ und $\beta^\vee$ müssen bzgl. G orthogonal sein. Hat ein 2-flat mindestens 3 Elemente α, β, $\gamma = a\alpha + b\beta$ mit den Eigenwerten λ, μ, ν bzgl. $G_\Pi G^{-1}$, so muss

$$G_\Pi G^{-1}\gamma = \nu\gamma = \lambda a\alpha + \mu b\beta \tag{2.14}$$

gelten, so dass alle Eigenwerte gleich sein müssen. Das bedeutet, dass $G_\Pi G^{-1}$, einge-

schränkt auf die von Π aufgespannt Ebene π, proportional zur Identität sein muss,

$$G_\Pi G^{-1}|_\pi = \lambda \mathbf{1}|_\pi \quad \forall\, \Pi \,\text{mit}\, |\Pi| > 2. \tag{2.15}$$

Äquivalent dazu ist die (2.10) entsprechende Bedingung

$$G_\Pi|_{\pi^\vee} = \lambda G|_{\pi^\vee} \quad \forall\, \Pi \,\text{mit}\, |\Pi| > 2 \tag{2.16}$$

für Kovektoren aus der Ebene $\pi^\vee = G^{-1}\pi$, welche besagt, dass G_Π und G, eingeschränkt auf $\pi^\vee$ proportional sein müssen.

Bezeichne nun $\hat{\pi}$ einen Projektor in V^*, der auf die Ebene π projiziert. Damit kann man die $\vee$-Bedingungen schreiben als

$$\begin{cases} \alpha G^{-1}\beta = 0 & \forall\, \Pi = \{\alpha, \beta\} \\ G_\Pi G^{-1}\hat{\pi} = \lambda\hat{\pi} & \forall\, \Pi \,\text{mit}\, |\Pi| > 2 \end{cases}. \tag{2.17}$$

Da man aufgrund der $GL(n)$-Invarianz der WDVV-Gleichungen jede Kovektor-Lösung in eine mit $G = \boldsymbol{1}$ transformieren kann (z.B. durch eine Cholesky-Zerlegung), bzw. umgekehrt alle Kovektor-Lösungen aus denen mit $G = \boldsymbol{1}$ erhalten kann, reicht es, die Lösungen mit $G = \boldsymbol{1}$ zu finden. Für diesen Fall ist aber auch die „$\vee$"-Abbildung die Identität und es ist sinnvoll, V mit V^* zu identifizieren. Die $\vee$-Bedingungen vereinfachen sich dann zu

$$G_\Pi \alpha = \lambda\alpha \quad \forall\, (\alpha, \Pi) \,\text{mit}\, \alpha \in \Pi, \quad \lambda = \lambda(\alpha, \Pi) \in \mathbb{C} \tag{2.18}$$

bzw.

$$\begin{cases} \alpha \cdot \beta = 0 & \forall\, \Pi = \{\alpha, \beta\} \\ G_\Pi|_\pi = \lambda \mathbf{1}|_\pi \,\text{mit}\, \lambda = \lambda(\Pi) \in \mathbb{C} & \forall\, \Pi \,\text{mit}\, |\Pi| > 2 \end{cases}. \tag{2.19}$$

Die Bedingungen für die Ebenen mit mehr als 2 Kovektoren kann man auch schreiben als

$$G_\Pi = \lambda\hat{\pi} \quad \text{mit}\, \lambda = \lambda(\Pi) \in \mathbb{C} \tag{2.20}$$

oder in 2-dimensionalen kartesischen Koordinaten der *jeweiligen Ebene* als

$$\sum_{\alpha\in\Pi} \alpha_i\alpha_j = \lambda\delta_{ij}, \quad i, j \in \{1, 2\}. \tag{2.21}$$

Im Reellen entspricht dies je 2 Gleichungen für die Kovektoren der Ebene, für die sich in Polarkoordinaten $(\alpha_1, \alpha_2) = (r_\alpha \cos\varphi_\alpha,\, r_\alpha \sin\varphi_\alpha)$

$$\sum_\alpha \alpha_1\alpha_1 = \sum_\alpha \alpha_2\alpha_2 \Longleftrightarrow \sum_\alpha r_\alpha^2 \cos^2\varphi_\alpha = \sum_\alpha r_\alpha^2 \sin^2\varphi_\alpha \Longleftrightarrow \sum_\alpha r_\alpha^2 \cos(2\varphi_\alpha) = 0, \tag{2.22}$$

$$\sum_\alpha \alpha_1\alpha_2 = 0 \Longleftrightarrow \sum_\alpha r_\alpha^2 \cos\varphi_\alpha \sin\varphi_\alpha = 0 \Longleftrightarrow \sum_\alpha r_\alpha^2 \sin(2\varphi_\alpha) = 0 \tag{2.23}$$

ergibt. Damit lassen sich die $\vee$-Bedingungen durch die Längen der Kovektoren und die

Winkel zwischen ihnen ausdrücken,

$$\begin{cases} \sphericalangle(\alpha,\beta) = 90^\circ & \forall\, \Pi = \{\alpha,\beta\} \\ \sum\limits_{\alpha\in\Pi} r_\alpha^2 \cos(2\varphi_\alpha^{(\Pi)}) = 0,\ \sum\limits_{\alpha\in\Pi} r_\alpha^2 \sin(2\varphi_\alpha^{(\Pi)}) = 0 & \forall\, \Pi \,\text{mit}\, |\Pi| > 2 \cdot \end{cases} \tag{2.24}$$

r_α bezeichnet hier die Länge von α und $\varphi_\alpha^{(\Pi)}$ den Winkel zwischen α und einem beliebigen, aber für jedes 2-flat Π festen, „Referenz-Kovektor" in der jeweiligen Ebene π.

2.1.3 Äquivalenz der $\vee$-Bedingungen zu den WDVV-Gleichungen

Die Äquivalenz der $\vee$-Bedingungen zu den WDVV-Gleichungen wurde von A. P. Veselov in mehreren Schritten in [29], [30], [34] bewiesen, hier wird der Beweis insgesamt dargestellt, für den komplexen Fall (2.1), der den reellen Ansatz (2.3) einschließt.

Wir gehen aus von den WDVV-Gleichungen in der Form (1.25),

$$F_i G^{-1} F_j = F_j G^{-1} F_i, \quad i,j = 1,...,n \tag{2.25}$$

mit den $n \times n$-Matrizen dritter Ableitungen

$$(F_i)_{kl} = \partial_i \partial_k \partial_l F \tag{2.26}$$

und der Metrik

$$G = \frac{1}{4} x^i F_i. \tag{2.27}$$

Mit der Definition

$$F_a := \frac{1}{4} a^i F_i \tag{2.28}$$

für einen Vektor $a \subset V$ ist (2.25) äquivalent zu

$$F_a G^{-1} F_b = F_b G^{-1} F_a \quad \forall a,b \in V \tag{2.29}$$

wie man sofort durch Wahl von $a = e_i$, $b = e_j$ in (2.29) bzw. bilden der Kontraktionen von (2.25) mit $a_i b_j$ sieht.

Für den Kovektor-Ansatz (2.1) haben die dritten Ableitungen die einfache Form

$$\partial_i \partial_k \partial_l F(x) = 4 \frac{\alpha_i \alpha_k \alpha_l}{\alpha(x)}, \tag{2.30}$$

somit wird

$$F_a = \sum_{\alpha\in\mathcal{A}} \frac{\alpha(a)}{\alpha(x)} \alpha \otimes \alpha \tag{2.31}$$

und G wird unabhängig von x und entspricht der Definition in den $\vee$-Bedingungen (2.6),

$$G = \sum_{\alpha\in\mathcal{A}} \alpha \otimes \alpha$$

Damit folgt

$$F_a G^{-1} F_b = \sum_{\alpha,\beta\in\mathcal{A}} \frac{\alpha(a)\beta(b)}{\alpha(x)\beta(x)} \alpha(\beta^\vee)\alpha\otimes\beta \tag{2.32}$$

und (2.29) wird zu

$$\sum_{\alpha,\beta\in\mathcal{A}} \frac{\alpha(\beta^\vee)}{\alpha(x)\beta(x)} (\alpha(a)\beta(b) - \alpha(b)\beta(a))\,\alpha\otimes\beta = 0 \quad \forall a,b\in V,\ \forall x\in V. \tag{2.33}$$

Wegen

$$\sum_{\alpha,\beta} \alpha(a)\beta(b)\,\alpha\otimes\beta = \sum_{\alpha,\beta} \beta(a)\alpha(b)\,\beta\otimes\alpha \tag{2.34}$$

kann man (2.33) auch schreiben als

$$\sum_{\alpha,\beta\in\mathcal{A}} \frac{\alpha(\beta^\vee)}{\alpha(x)\beta(x)} (\alpha(a)\beta(b) - \alpha(b)\beta(a))(\alpha\otimes\beta - \beta\otimes\alpha) = 0 \quad \forall a,b\in V,\ \forall x\in V \tag{2.35}$$

oder mit

$$(\alpha\otimes\beta - \beta\otimes\alpha)\otimes(\alpha\otimes\beta - \beta\otimes\alpha) =: (\alpha\wedge\beta)^{\otimes 2} \tag{2.36}$$

einheitlicher als

$$\sum_{\alpha,\beta\in\mathcal{A}} \frac{\alpha(\beta^\vee)}{\alpha(x)\beta(x)} (\alpha\wedge\beta)^{\otimes 2} = 0 \quad \forall x\in V. \tag{2.37}$$

Dass die $\vee$-Bedingungen (2.37) implizieren ist nun schnell zu beweisen (Theorem 1 in [29]): Wenn $\mathcal{A}$ die $\vee$-Bedingungen erfüllt verschwinden in (2.37) bereits die Teilsummen über die Elemente der 2-flats Π einzeln,

- für 2-elementige $\Pi = \{\alpha,\beta\}$ verschwindet der einzige Summand wegen $\alpha(\beta^\vee) = 0$ (was in diesem Fall die $\vee$-Bedingung ist)
- für 2-flats mit mehr als 2 Elementen ist die Teilsumme proportional zu der Relation (2.37) für die Funktion

$$F_\Pi\big|_{\pi^\vee} = \sum_{\alpha\in\Pi} \alpha(x)^2 \log\alpha(x)^2\Big|_{x\in\pi^\vee}. \tag{2.38}$$

  Diese ist aber erfüllt, da $\pi^\vee$ 2-dimensional ist und in 2 Dimensionen die WDVV-Gleichungen von jeder Funktion gelöst werden[4].

Jetzt müssen wir noch beweisen, dass aus (2.37) die $\vee$-Bedingungen folgen. Dies wurde in [30] für den reellen Fall (teils sehr knapp) gezeigt und in [34] auf den komplexen Fall verallgemeinert (und der Beweis dabei teilweise vereinfacht).

Die Grenzwertbetrachtung

$$x\to x',\ \alpha(x') = 0 \tag{2.39}$$

[4] $F_iF_k^{-1}F_j = F_jF_k^{-1}F_i\ \forall i,j,k\in\{1,2\} \Longleftrightarrow F_1F_1^{-1}F_2 = F_2F_1^{-1}F_1 \ \vee\ F_1F_2^{-1}F_2 = F_2F_2^{-1}F_1$

liefert zunächst (2.37) $\Longrightarrow$

$$\sum_{\beta\in\mathcal{A}} \frac{\alpha(\beta^\vee)}{\beta(x)}(\alpha\wedge\beta)^{\otimes 2}\Big|_{\alpha(x)=0} \equiv 0 \quad \forall\alpha\in\mathcal{A}. \tag{2.40}$$

Die Summanden in (2.40), bei denen β in einer Ebene Π mit α liegt kann man alle schreiben als

$$\frac{\alpha((\lambda\beta_\Pi+\mu\alpha)^\vee)}{(\lambda\beta_\Pi+\mu\alpha)(x)}(\alpha\wedge(\lambda\beta_\Pi+\mu\alpha))^{\otimes 2}\Big|_{\alpha(x)=0} = \frac{\lambda\alpha((\lambda\beta_\Pi+\mu\alpha)^\vee)}{\beta_\Pi(x)}(\alpha\wedge\beta_\Pi)^{\otimes 2}\Big|_{\alpha(x)=0} \tag{2.41}$$

mit einem Repräsentanten $\beta_\Pi\in\Pi$, woraus man erkennt, dass sie die selben Singularitäten haben. Somit kann $\frac{(\alpha\wedge\beta_\Pi)^{\otimes 2}}{\beta_\Pi(x)}$ aus diesen Teilsummen ausgeklammert werden und man erhält (2.40) $\Longleftrightarrow$

$$\sum_{\Pi\ni\alpha} \frac{\lambda_\Pi\cdot(\alpha\wedge\beta_\Pi)^{\otimes 2}}{\beta_\Pi(x)}\Big|_{\alpha(x)=0} \equiv 0 \quad \forall\alpha\in\mathcal{A} \tag{2.42}$$

mit den Koeffizienten

$$\lambda_\Pi(\alpha,\Pi) = \sum_{\lambda,\mu}\lambda\alpha((\lambda\beta_\Pi+\mu\alpha)^\vee). \tag{2.43}$$

Die $\beta_\Pi|_{\alpha(x)=0}$ sind nun aber alle verschieden, so dass die Summanden in (2.42) bereits einzeln verschwinden müssen, also (2.40) $\Longleftrightarrow$

$$\sum_{\beta\in\Pi} \frac{\alpha(\beta^\vee)}{\beta(x)}(\alpha\wedge\beta)^{\otimes 2}\Big|_{\alpha(x)=0} \equiv 0 \quad \forall\,\text{Paare}\,(\alpha,\Pi)\,\text{mit}\,\alpha\in\Pi \tag{2.44}$$

Aus (2.44) folgt nun

$$\sum_{\beta\in\Pi}\alpha(\beta^\vee)\alpha\wedge\beta = 0 \quad \forall\,(\alpha,\Pi)\,\text{mit}\,\alpha\in\Pi \tag{2.45}$$

durch einsetzen von $x\otimes y$ mit einem $y\in V$, $\alpha(y)\neq 0$. Es ist nämlich

$$(\alpha\wedge\beta)(x\otimes y) = -\beta(x)\alpha(y) \tag{2.46}$$

wegen $\alpha(x)=0$, so dass sich in (2.44) $\beta(x)$ aus dem Bruch kürzt und $\alpha(y)$ aus der Gleichung und man (2.45) erhält.

(2.45) bedeutet

$$\sum_{\beta\in\Pi}\alpha(\beta^\vee)(\alpha(a)\beta(b)-\alpha(b)\beta(a)) = 0 \quad \forall a,b\in V,\ \forall\,(\alpha,\Pi)\,\text{mit}\,\alpha\in\Pi, \tag{2.47}$$

was mit der Definition von G_Π aus (2.6), $G_\Pi = \sum_{\beta\in\Pi}\beta\otimes\beta$ wegen $\alpha(\beta^\vee) = \beta(\alpha^\vee)$

geschrieben werden kann als

$$\alpha(a)G_\Pi(\alpha^\vee, b) = \alpha(b)G_\Pi(\alpha^\vee, a) \quad \forall a, b \in V, \ \ \forall\, (\alpha, \Pi) \,\text{mit}\, \alpha \in \Pi. \tag{2.48}$$

Hieran sieht man, dass das Verhältnis $G_\Pi(\alpha^\vee, a)/\alpha(a)$ nicht von dem Vektor $a \in V$ abhängt,

$$G_\Pi(\alpha^\vee, a) = \lambda\alpha(a) \quad \forall a \in V, \ \ \forall\, (\alpha, \Pi) \,\text{mit}\, \alpha \in \Pi. \tag{2.49}$$

Dies bedeutet aber

$$G_\Pi G^{-1}\alpha = \lambda\alpha \quad \forall\, (\alpha, \Pi) \,\text{mit}\, \alpha \in \Pi, \tag{2.50}$$

was die $\vee$-Bedingungen (2.11) sind. □

2.2 Auffinden von $\vee$-Systemen

2.2.1 Coxeter-Wurzelsysteme

Eine bedeutende Klasse von $\vee$-Systemen liefern direkt die Coxeter-Wurzelsysteme [35], wie A. P. Veselov 1999 gezeigt hat [29]. Sie sind definiert wie folgt.

Sei V eine Euklidischer Vektorraum mit einem Skalarprodukt, und sei $\mathcal{G}$ sei eine *Coxeter-Gruppe*, das ist eine irreduzible, endliche Gruppe, die durch orthogonale Reflexionen

$$s_\alpha(\beta) := \beta - \frac{2\langle\alpha,\beta\rangle}{\langle\alpha,\alpha\rangle}\alpha \tag{2.51}$$

an Hyperebenen in V erzeugt wird. Sei $\mathcal{R}$ eine Menge von Normalenvektoren zu den Reflexions-Hyperebenen von $\mathcal{G}$. Die Länge dieser Normalenvektoren sei so gewählt, dass $\mathcal{R}$ invariant unter der natürlichen Wirkung von $\mathcal{G}$ ist und für jede Hyperebene genau zwei Normalenvektoren enthält. Wir wählen nun von jedem solchen Paar einen aus (z.B. immer den, der positiv bzgl. einer bestimmten Linearform über V ist) und bilden daraus das System $\mathcal{R}_+$,

$$\mathcal{R} = \mathcal{R}_+ \cup (-\mathcal{R}_+) \tag{2.52}$$

Solch ein System $\mathcal{R}_+$ ist ein *Coxeter-Wurzelsystem* und seine Elemente heißen *Wurzeln*.

Jedes Coxeter-Wurzelsystem $\mathcal{R}_+$ ist ein $\vee$-System, wie mit den $\vee$-Bedingungen einfach zu beweisen ist: Die Form

$$G = \sum_{\alpha\in\mathcal{R}_+} \alpha\otimes\alpha \tag{2.53}$$

ist proportional zur euklidischen Struktur von V, da sie invariant unter $\mathcal{G}$ ist und $\mathcal{G}$ irreduzibel ist. Dies ist aber auch der Fall für

$$G_\pi = \sum_{\alpha\in\mathcal{R}_+\cap\pi} \alpha\otimes\alpha \tag{2.54}$$

für 2-Ebenen π, die mehr als zwei Wurzeln von $\mathcal{R}_+$ enthalten, so dass die $\vee$-Bedingungen (2.16) für diese Ebenen erfüllt sind. Enthält eine Ebene π nur zwei Wurzeln, so müssen diese orthogonal sein, was die $\vee$-Bedingung für diesen Fall ist. □

2.2.2 Projektionen auf Unterräume

Eine sehr erfolgreiche Methode, um aus bekannten $\vee$-Systemen neue zu erhalten, ist die Projektion auf bestimmte Unterräume.

Sei $\mathcal{A}$ ein $\vee$-System im Vektorraum V^*, den wir hier mit V identifizieren. Weiter sei $\mathcal{B}\subset\mathcal{A}$ der Schnitt von $\mathcal{A}$ mit einem Unterraum U' von $V = V^*$, und U sei der von $\mathcal{B}$ aufgespannte Unterraum. Sei L das orthogonale Komplement von U in V,

$$L = \bigcap_{\beta\in\mathcal{B}} \{x\in V:\ \beta(x) = 0\}. \tag{2.55}$$

Dann erhält man ein $\vee$-System $\mathcal{A}'$ in L wie folgt:

- Die Elemente $\alpha \in \mathcal{B}$ werden entfernt
- Die Elemente $\alpha \in \mathcal{A} \cap L$ werden übernommen ($\mathcal{A} \cap L$ kann leer sein)
- Die restlichen $\alpha \in \mathcal{A}$ werden orthogonal auf L projiziert. Entstehen dabei kollineare Mengen $\{\gamma_i = \lambda_i \gamma\}_{i=1,\dots,k}$ von Kovektoren, werden diese je durch einen dazu für die Definition (2.1) von F äquivalenten Kovektor $\tilde{\gamma} = \lambda\gamma$ mit $\lambda^2 = \sum_{i=1}^{k} \lambda_i^2$ ersetzt

Man beachte, dass die vom System $\mathcal{A}$ definierte Kovektor-Lösung (2.1) $F_{\mathcal{A}}$ zunächst nur im Komplement

$$\tilde{V} = V \setminus \bigcup_{\alpha \in \mathcal{A}} \{x \in V : \alpha(x) = 0\} \tag{2.56}$$

von V zur Vereinigung der zu den Elementen von $\mathcal{A}$ orthogonalen Hyperebenen definiert ist, aber auf ganz V analytisch fortgesetzt werden kann. Die vom projizierten System $\mathcal{A}'$ in L definierte Funktion $F_{\mathcal{A}'}$ ist somit in L identisch zu der analytischen Fortsetzung von $F_{\mathcal{A}}$ nach L,

$$F_{\mathcal{A}}\big|_L = F_{\mathcal{A}'}, \tag{2.57}$$

weshalb man auch von Einschränkung oder Restriktion spricht.

Dass das so erzeugte System $\mathcal{A}'$ tatsächlich wieder ein $\vee$-System ist, bzw. dass $F_{\mathcal{A}'}$ die WDVV-Gleichungen *in* L löst, wurde von M. V. Feigin und A. P. Veselov 2005 in [33] gezeigt, im Zusammenhang mit den Frobenius-Strukturen, welche schon 1992 von B. Dubrovin mit den WDVV-Gleichungen in Verbindung gebracht wurden [3]. Der Beweis benutzt die tieferen Zusammenhänge jedoch nicht, wir brauchen nur folgende Feststellung:

Definiert man für ein $\vee$-System $\mathcal{A}$, bzw. die entsprechende Kovektor-Lösung F, eine Multiplikation „$*$" auf dem Tangentialbündel von V als

$$u * v = G^{-1} F_u v = G^{-1} F_v u = \sum_{\alpha \in \mathcal{A}} \frac{\alpha(u)\alpha(v)}{\alpha(x)} \alpha^{\vee} \tag{2.58}$$

mit F_u wie in (2.28), so entspricht ihre Assoziativität gerade den WDVV-Gleichungen, wie man an (2.29) direkt sieht. Diese Multiplikation ist zunächst auch nur für $x \in \tilde{V}$ definiert

Sei nun $\mathcal{A}$ ein $\vee$-System, so dass das entsprechende „$*$"-Produkt auf dem Tangentialbündel von V assoziativ ist. $\mathcal{B}$ sei wie oben der Schnitt von$\mathcal{A}$ mit einem Unterraum, und L der durch (2.55) definierte Unterraum, auf den $\mathcal{A}$ projiziert wird. Weiter sei $\mathcal{C} = \mathcal{A} \backslash \mathcal{B}$ und

$$\tilde{L} = L \setminus \bigcup_{\gamma \in \mathcal{C}} \{x \in V : \gamma(x) = 0\}. \tag{2.59}$$

Wir werden nun zeigen, dass die „$*$"-Multiplikation auf $\tilde{L}$ fortgesetzt werden kann und $\tilde{L}$ unter ihr abgeschlossen ist, so dass nun rückwärts ihre Assoziativität die Gültigkeit der WDVV-Gleichungen in L für $F_{\mathcal{A}'} = F_{\mathcal{A}}\big|_L$ impliziert.

Dazu betrachten wir einen Punkt $x_0 \in \tilde{L}$ und zwei Tangentialvektoren u, v in x_0, die tangential zu L sind und setzten sie zu lokal analytischen Vektorfeldern $u(x)$, $v(x)$ im

gesamten Vektorraum V fort. Außerhalb von L ist dann die „$*$"-Multiplikation wohldefiniert, die Frage ist, was für $x \to x_0$ passiert.

Dazu betrachten wir $u(x) * v(x)$ in der Nähe einer Hyperebene

$$\pi_\beta = \{x \in V : \beta(x) = 0\} \tag{2.60}$$

für ein $\beta \in \mathcal{B}$. In euklidischen lokalen Koordinaten $(t, \boldsymbol{s})$ mit der Koordinaten $t = \beta(x)$ senkrecht zu π_β und dem Vektor $\boldsymbol{s}$ der $n-1$ Koordinaten der orthogonalen Projektion von x auf π_β. Die Vektorfelder können damit geschrieben werden als

$$\begin{aligned} u(x) &= u(t, \boldsymbol{s}) = a(t, \boldsymbol{s})\partial_t + \xi(t, \boldsymbol{s}), \\ v(x) &= v(t, \boldsymbol{s}) = b(t, \boldsymbol{s})\partial_t + \eta(t, \boldsymbol{s}), \end{aligned} \tag{2.61}$$

mit zu π_β parallelen Vektorfeldern ξ und η, $\beta(\xi) = \beta(\eta) = 0$. Da $u(x)$ und $v(x)$ in x_0 nun tangential zu π_β sind, ist $a(0, \boldsymbol{s}) = b(0, \boldsymbol{s}) = 0$ und aus der vorausgesetzten Analytiziät folgt

$$\lim_{t \to 0} \frac{a(t, \boldsymbol{s})b(t, \boldsymbol{s})}{t} = 0. \tag{2.62}$$

Das heißt, dass der β-Term in der Summe (2.58) von $u(x) * v(x)$ bei $\beta(x) = 0$ verschwindet. Diese Folgerung gilt aber für alle $\beta \in \mathcal{B}$, so dass

$$\lim_{x \to x_0} u(x) * v(x) = \sum_{\alpha \in \mathcal{A} \setminus B} \frac{\alpha(u)\alpha(v)}{\alpha(x_0)} \alpha^\vee \tag{2.63}$$

und das „$*$"-Produkt auch in $\tilde{L}$ wohldefiniert ist.

Es bleibt noch zu zeigen, dass $u * v$ tangential zu $\tilde{L}$ ist, sofern u und v es sind. Dies folgt wieder aus der Betrachtung für die einzelnen Hyperebenen π_β senkrecht zu den $\beta \in \mathcal{B}$. Für diese ist zu zeigen, dass

$$\beta(u * v) = \sum_{\alpha \in \mathcal{A} \setminus \{\beta\}} \frac{\alpha(u)\alpha(v)}{\alpha(x)} \beta(\alpha^\vee) = 0 \tag{2.64}$$

für $x \in \pi_\beta$, $u, v \in T_x\pi_\beta$. Aus den WDVV-Gleichungen für die von $\mathcal{A}$ definierte Funktion F in der Form (2.40),

$$\sum_{\beta \in \mathcal{A}} \frac{\alpha(\beta^\vee)}{\beta(x)} (\alpha \wedge \beta)^{\otimes 2}\Big|_{\alpha(x)=0} = 0 \quad \forall \alpha \in \mathcal{A} \tag{2.65}$$

erhält man durch einsetzten

$$\sum_{\alpha \in \mathcal{A}} \frac{\alpha(\beta^\vee)}{\alpha(x)} (\alpha(a)\beta(b) - \alpha(b)\beta(a))(\alpha(z)\beta(y) - \alpha(y)\beta(z)) \equiv 0\Big|_{\alpha(x)=0} \tag{2.66}$$

für beliebige Vektoren a, b, y, $z \in V$. Für $b = u \in \pi_\beta$, $y = v \in \pi_\beta$ wird $\beta(b) = \beta(y) = 0$,

und wir erhalten

$$\beta(a)\beta(z)\sum_{\alpha\in\mathcal{A}\setminus\{\beta\}}\frac{\alpha(\beta^\vee)}{\alpha(x)}\alpha(u)\alpha(v)=0, \tag{2.67}$$

was für a, z mit $\beta(a)\neq 0$, $\beta(z)\neq 0$ (2.64) impliziert. □

2.2.3 Verallgemeinerte Wurzelsysteme

Eine weitere Klasse von $\vee$-Systemen liefern Deformationen der von V. Serganova 1996 in [36] im Zusammenhang mit klassischen Lie-Superalgebren eingeführten verallgemeinerten Wurzelsysteme. Sie sind definiert wie folgt.

Sei V ein endlich dimensionaler komplexer Vektorraum mit einer nicht degeneriertem Bilinearform $\langle\cdot,\cdot\rangle$. Eine endliche Menge $R\subset V\setminus\{0\}$ ist ein *verallgemeinertes Wurzelsystem*, wenn folgende Bedingungen erfüllt sind:

- R spannt V auf und $R=-R$
- für $\alpha,\beta\in R$, $\langle\alpha,\alpha\rangle\neq 0$ gilt $\quad\frac{2\langle\alpha,\beta\rangle}{\langle\alpha,\alpha\rangle}\in\mathbb{Z}$, $s_\alpha(\beta):=\beta-\frac{2\langle\alpha,\beta\rangle}{\langle\alpha,\alpha\rangle}\alpha\in R$
- für $\alpha,\beta\in R$, $\langle\alpha,\alpha\rangle=0$, $\langle\alpha,\beta\rangle\neq 0$ gehört wenigstens einer der Vektoren $\alpha+\beta$ oder $\alpha-\beta$ zu R

Jedes verallgemeinerte Wurzelsystem besitzt eine partielle Symmetrie, beschrieben durch die endliche Gruppe W_0, welche durch die Reflexionen an den nicht-isotropen Wurzeln erzeugt wird.

Wir betrachten nun Deformationen dieser Systeme, die von A. N. Sergeev und A. P. Veselov 2003 in [37], im Zusammenhang mit Deformationen der Lie-Superalgebren und von Calogero-Moser Systemen, eingeführt wurden. Sie werden *zulässige Deformationen* genannt. Sie sind definiert über eine Deformation der Bilinearform $\langle\cdot,\cdot\rangle$ (parametrisiert über ein k, so dass $k=-1$ dem undeformierten System entspricht), und einer Skalierung der Wurzeln $\alpha\in R$, $\alpha\to m_\alpha\alpha$. Eine Deformation ist *zulässig,* wenn folgende Bedingungen erfüllt sind:

- Die deformierte Bilinearform $\langle\cdot,\cdot\rangle_k$ und die Skalierungen m_α sind invariant unter Wirkung der Gruppe W_0 der partielle Symmetrie
- die isotropen Wurzeln werden nicht skaliert
- die Funktion $\psi_0=\prod_{\alpha\in R_+}\sin^{-m_\alpha}(\alpha,x)$ ist eine (formale) Eigenfunktion des *Schrödingeroperators*

$$L=-\Delta+\sum_{\alpha\in R_+}\frac{m_\alpha(m_\alpha+2m_{2\alpha}+1)\langle\alpha,\alpha\rangle_k}{\sin^2\langle\alpha,x\rangle_k} \tag{2.68}$$

wobei sich der Laplace-Operator Δ auf die deformierte Bilinearform $\langle\cdot,\cdot\rangle_k$ bezieht, welche als nicht degeneriert vorausgesetzt wird

Für jede zulässige Deformation $(R,\, k,\, \{m_\alpha\})$ eines verallgemeinerten Wurzelsystems R ist nun die Menge $\mathcal{A} = \{\sqrt{m_\alpha}\alpha,\, \alpha \in R\}$ ein $\vee$-System, wenn die kanonische Form

$$G_{\mathcal{A}}(u, v) = \sum_{\alpha \in \mathcal{A}} m_\alpha \alpha(u)\alpha(v) \tag{2.69}$$

nicht degeneriert ist, wie M. V. Feigin und A. P. Veselov 2007 in [34] gezeigt haben.

3 Klassifizierung und Konstruktion von Kovektor-Lösungen mit Hypergraphen

Die Algorithmen wurden der Einfachheit halber zunächst in dem bei mathematischen Physikern beliebten und sehr mächtigen, aber proprietären Computer-Algebra-System *Mathematica* (Version 7.0.1) der Firma Wolfram Research [41] implementiert. Für die Weiterentwicklung der Algorithmen schlage ich jedoch die Verwendung einer standardisierten Programmiersprache vor, für die freie Compiler/Interpreter vorliegen.

Da die Algorithmen zur Konstruktion der $\vee$-Systeme keine Funktionen eines Computer-Algebra-Systems benötigen, die man nicht einfach selbst implementieren könnte, bietet sich eine effiziente allgemeine Programmiersprache wie C++ an. Sollten Funktionen eines Computer-Algebra-Systems benötigt werden, schlage ich die Verwendung von freien Systemen wie Axiom [42] oder Maxima [43] vor, so dass alle von Computern ausgeführten Rechenschritte nachvollziehbar sind, wodurch die Ergebnisse erst wirklich verlässlich werden.

3.1 Analyse von gegebenen $\vee$-Systemen

`all2Flats[A]` gibt eine Liste `Ids` aller 2-flats des Systems, dessen Vektoren in der Matrix `A` stehen, jedes Element von `Ids` ist eine Liste der Indizes der in dem jeweiligen 2-flat enthaltenen Vektoren,

```
all2Flats[A_] := Block[{At, n, Ids},
   At = Transpose[A]; n = Length[At]; Ids = {{}};
   Do[If[! pairListedQ[Ids, i, j], AppendTo[Ids, {i, j}];
     Do[If[MatrixRank[{At[[i]], At[[j]], At[[k]]}] == 2, AppendTo[Ids[[Length[Ids]]], k]],
      {k, j + 1, n}]],
    {i, 1, n - 1}, {j, i + 1, n}];
   Ids[[2 ;;]]
  ];
```

hier wird

```
pairListedQ[L_, i_, j_] :=
  Catch[If[i == j, Throw[False]]; Do[If[MemberQ[l, i] && MemberQ[l, j], Throw[True]], {l, L}];
   False];
```

benutzt.

`large2Flats[A]` gibt eine Liste aller 2-flats mit mehr als 2 Elementen des Systems `A`

```
large2Flats[A_] := Select[all2Flats[A], Length[#] > 2 &];
```

`addSFs[lFlats]` fügt die 2-flats mit 2 Elementen hinzu,

```
addSFs[lFlats_] := Block[{allFlats = lFlats, lbl},
   Do[Do[If[Sort[lf ⋂ pair] == Sort[pair], Goto[lbl]], {lf, lFlats}];
    AppendTo[allFlats, pair]; Label[lbl],
    {pair, Subsets[DeleteDuplicates[Flatten[lFlats]], {2}]}];
   Return[allFlats]
  ];
```

,

und `remSFs` entfernt sie,

```
remSFs[allFlats_] := Select[allFlats, Length[#1] > 2 &];
```

`checkV[A]` überprüft die ∨-Bedingungen in der Form (2.12) für das System `A`.

```
checkV[A_] := Block[{At, G, Gi, mults, m, result},
   At = Transpose[A]; G = A.At; Gi = Inverse[G]; result = {};
   Do[mults = {};
    Do[m = Solve[Sum[At[[i]].Gi.At[[id]] * At[[i]], {i, ids}] == l * At[[id]], l];
     AppendTo[mults, If[m == {}, False, Simplify[m[[1, 1, 2]]]]],
     {id, ids}];
    AppendTo[result, {ids,
      If[Do[If[m =!= mults[[1]], Return[True]], {m, mults}] === True, mults, mults[[1]]]}],
    {ids, all2Flats[A]}];
   Return[result]
  ];
```

,

`checkVH[A]` überprüft die ∨-Bedingungen für eine flache Metrik in der Darstellung 2.3 mit den Vektoren `vs` und den Mulfiplizitäten `ms`,

```
checkVH[flats_, vs_, ms_] := Block[{mults, m, result},
   result = {};
   Do[mults = {};
    Do[m = Solve[Sum[ms[[i]] * Sqrt[ms[[id]]] * vs[[i]].vs[[id]] * vs[[i]], {i, ids}]
        == l * Sqrt[ms[[id]]] * vs[[id]], l];
     AppendTo[mults, If[m == {}, False, Simplify[m[[1, 1, 2]]]]],
     {id, ids}];
    AppendTo[result, {ids,
      If[Do[If[m =!= mults[[1]], Return[True]], {m, mults}] === True, mults, mults[[1]]]}],
    {ids, flats}];
   Return[result]
  ];
```

3.2 Konstruktion „orthogonaler" Hypergraphen

Es gibt eine notwendige Eigenschaft für die Hypergraphen dafür, dass sie als Vektorsysteme realisiert werden können, welche die ∨-Bedingung für 2-flats mit 2 Elementen, die Orthogonalitätsbedingung, erfüllen:

Für jedes Paar aus einem 2-flat $\Pi = \{\beta\}$ und einem nicht in Π liegenden Vektor α gilt, dass α entweder auf allen β's senkrecht steht oder auf höchstens einem β. Eine Ausnahme tritt auf für 2-flats mit 4 oder mehr Elementen, *in* welchen zwei Vektoren orthogonal sind.

Die folgenden Algorithmen konstruieren Hypergraphen mit dieser „Orthogonalitätseigenschaft", ich nenne sie „orthogonale" Hypergraphen. Diese Eigenschaft schränkt die Menge der in Frage kommenden Hypergraphen außerordentlich ein.

3.2.1 Isomorphie von Hypergraphen

Entscheidend bei der Konstruktion der Hypergraphen ist die Beachtung ihre Isomormphien.

`edgesIsoQ[edgeList, edge1, edge2]` prüft für einen Hypergraphen, welcher als Liste seiner Kanten `edgeList` gegeben ist, ob die Kanten `edge1` und `edge2` isomprph sind, d.h. ob der Hypergraph von `edge1` und `edge2` aus betrachtet gleich aussieht.

```
edgesIsoQ[edgeList_, edge1_, edge2_] := Block[
   {vCount, eCount, vList, eList, e1 = edge1, e2 = edge2, eMap, vMap, eOrigins,
    vOrigins, eOrigins2, vOrigins2, levels, levels2, lLists, lLists2, levelsMaps,
    vHashes, eHashes, vMaxSize, eMaxSize, eRHashes, vRHashes, eRHashes2, vRHashes2,
    vCHashes, eCHashes, vCHashes2, eCHashes2, result},

   eList = edgeList; vCount = Max[Flatten[eList]]; eCount = Length[eList];
   vList = Table[{}, {vCount}]; Do[Do[AppendTo[vList[[v]], e], {v, eList[[e]]}], {e, eCount}];
   eMap = Table[0, {eCount}]; vMap = Table[0, {vCount}];
   eOrigins = Table[{}, {eCount}]; vOrigins = Table[{}, {vCount}];
   eOrigins2 = Table[{}, {eCount}]; vOrigins2 = Table[{}, {vCount}];
   eRHashes = Table[0, {eCount}]; vRHashes = Table[0, {vCount}];
   eRHashes2 = Table[0, {eCount}]; vRHashes2 = Table[0, {vCount}];
   vMaxSize = Max[Length /@ vList]; eMaxSize = Max[Length /@ eList];
   vHashes = Table[vHash[v, 2, vList, eList], {v, vCount}];
   eHashes = Table[eHash[e, 2, vList, eList], {e, eCount}];
   If[eHashes[[e1]] ≠ eHashes[[e2]], Return[False]];

   lLists = {{e1}}; vTreeC[lLists, vOrigins, eOrigins, vList, eList];
   If[Mod[Length[lLists], 2] == 0,
    vRevHashesC[Length[lLists], 1, lLists, vRHashes, eRHashes, vOrigins, eOrigins],
    eRevHashesC[Length[lLists], 1, lLists, vRHashes, eRHashes, vOrigins, eOrigins]];
   lLists2 = {{e2}}; vTreeC[lLists2, vOrigins2, eOrigins2, vList, eList];
   If[Mod[Length[lLists2], 2] == 0,
    vRevHashesC[Length[lLists2], 1, lLists2, vRHashes2, eRHashes2, vOrigins2, eOrigins2],
    eRevHashesC[Length[lLists2], 1, lLists2, vRHashes2, eRHashes2, vOrigins2, eOrigins2]];
   vCHashes = Table[vHashes[[v]] + Max[Length /@ vHashes] vRHashes[[v]], {v, vCount}];
   eCHashes = Table[eHashes[[e]] + Max[Length /@ eHashes] eRHashes[[e]], {e, eCount}];
   vCHashes2 = Table[vHashes[[v]] + Max[Length /@ vHashes] vRHashes2[[v]], {v, vCount}];
   eCHashes2 = Table[eHashes[[e]] + Max[Length /@ eHashes] eRHashes2[[e]], {e, eCount}];
   levels = {}; eGroupsC[]; levels2 = Table[{}, {Length[levels]}]; levels2[[1]] = {{e2}};
   levelsMaps = Table[{}, {Length[levels]}];
   Do[levelsMaps[[l]] = Table[{}, {Length[levels[[l]]]}], {l, Length[levels]}];

   eMap[[e2]] = e1; result = checkVLevel[2];
   If[result,
    Do[If[! Sort[Map[vMap[[#1]] &, eList, {2}][[e]]] == Sort[eList[[eMap[[e]]]]], Print["Error!"]],
     {e, eCount}]];
   Return[result];
  ];
```

Dazu wird der Hypergraph als normaler Graph betrachtet der als *seine* Vertices sowohl

die Kanten als auch Vertices des Hypergraphen enthält, immer abwechselnd verbunden durch neue Kanten. Der so erhaltene Graph wird nun als Baumstruktur von `edge1` als „Wurzel" aus und parallel dazu von `edge2` als „Wurzel" aus betrachtet. `vTreeC` und `eTreeC` werden dazu immer abwechselnd ausgeführt, um die „Ebenen" (`levels`) der Vertices dieser Bäume/Graphen aufzubauen,

```
SetAttributes[vTreeC, HoldAll];
vTreeC[lLists_, vOrigins_, eOrigins_, vList_, eList_] := Block[{list},
   list = {};
   Do[Do[If[! MemberQ[eOrigins[[e]], v], If[! MemberQ[list, v], AppendTo[list, v]];
      AppendTo[vOrigins[[v]], e]],
     {v, eList[[e]]}], {e, lLists[[-1]]}];
   If[list ≠ {}, AppendTo[lLists, list]; eTreeC[lLists, vOrigins, eOrigins, vList, eList],
    vOrigins = Sort /@ vOrigins; eOrigins = Sort /@ eOrigins];
  ];
SetAttributes[eTreeC, HoldAll];
eTreeC[lLists_, vOrigins_, eOrigins_, vList_, eList_] := Block[{list},
   list = {};
   Do[Do[If[! MemberQ[vOrigins[[v]], e], If[! MemberQ[list, e], AppendTo[list, e]];
      AppendTo[eOrigins[[e]], v]],
     {e, vList[[v]]}], {v, lLists[[-1]]}];
   If[list ≠ {}, AppendTo[lLists, list]; vTreeC[lLists, vOrigins, eOrigins, vList, eList],
    vOrigins = Sort /@ vOrigins; eOrigins = Sort /@ eOrigins];
  ];
```

Die Funktionen `vGroupsC` und `eGroupsC` gruppieren die Elemente der „Ebenen",

```
vGroupsC[] := Block[{},
   AppendTo[levels, Flatten[GatherBy[SortBy[lLists[[Length[levels] + 1]],
       {Sort[vOrigins[[#1]] &, vCHashes[[#1]] &}],
      {Sort[vOrigins[[#1]] &, vCHashes[[#1]] &}], 1]];
   If[Length[levels] < Length[lLists], eGroupsC[]]
  ];
eGroupsC[] := Block[{},
   AppendTo[levels, Flatten[GatherBy[SortBy[lLists[[Length[levels] + 1]],
       {Sort[eOrigins[[#1]] &, eCHashes[[#1]] &}],
      {Sort[eOrigins[[#1]] &, eCHashes[[#1]] &}], 1]];
   If[Length[levels] < Length[lLists], vGroupsC[]]
  ];
```

`checkVLevel` und `checkELevel` prüfen nun schließlich die „Ebenen" auf Übereinstimmung, `checkELevel` ist *vollständig* analog zu `checkVLevel` (`e` und `v` bzw. `E` und `V` sind vertauscht) und wird daher nicht angegeben,

```
checkVLevel[l_] := Block[{g, try, vMappedOrigins},
   Catch[
    vMappedOrigins = Sort /@ Map[eMap[[#1]] &, vOrigins2, {2}];
    levels2[[l]] = Flatten[GatherBy[SortBy[lLists2[[l]],
        {vMappedOrigins[[#1]] &, vCHashes2[[#1]] &}],
       {vMappedOrigins[[#1]] &, vCHashes2[[#1]] &}], 1];
    If[Map[Length, levels2[[l]], {1}] ≠ Map[Length, levels[[l]], {1}], Throw[False]];
    Do[If[vOrigins[[levels[[l]][[g]][[1]]] ≠ vMappedOrigins[[levels2[[l]][[g]][[1]]]
       || vCHashes[[levels[[l]][[g]][[1]]] ≠ vCHashes2[[levels2[[l]][[g]][[1]]], Throw[False]];
     levelsMaps[[l]][[g]] = Range[Length[levels[[l]][[g]]]];
     Do[vMap[[levels2[[l]][[g]][[v]]] = levels[[l]][[g]][[v]], {v, Length[levels2[[l]][[g]]]}],
     {g, Length[levels2[[l]]]}];
    If[l == Length[levels], Throw[True]];
    Label[try]; If[checkELevel[l + 1], Throw[True]];
    For[g = 1, g ≤ Length[levels[[l]]], g++,
     If[Length[levels[[l]][[g]]] > 1, levelsMaps[[l]][[g]] = NextPermutation[levelsMaps[[l]][[g]]];
      Do[vMap[[levels2[[l]][[g]][[v]]] = levels[[l]][[g]][[levelsMaps[[l]][[g]][[v]]],
       {v, Length[levels2[[l]][[g]]]}];
      If[levelsMaps[[l]][[g]] ≠ Range[Length[levels[[l]][[g]]]], Goto[try]]]];
    Throw[False]]
   ];
```

Zur Beschleunigung werden Hashes verwendet,

```
vHash[v_, l_, vList_, eList_] := Length[vList[[v]]] +
  If[l > 1, (1 + vMaxSize) * Sum[eHash[e, l - 1, vList, eList], {e, vList[[v]]}], 0];
eHash[e_, l_, vList_, eList_] := Length[eList[[e]]] +
  If[l > 1, (1 + eMaxSize) * Sum[vHash[v, l - 1, vList, eList], {v, eList[[e]]}], 0];

SetAttributes[vRevHashesC, HoldAll];
vRevHashesC[l_, m_, lLists_, vRHashes_, eRHashes_, vOrigins_, eOrigins_] := Block[{},
   Do[vRHashes[[v]] += m; Do[eRHashes[[e]] += vRHashes[[v]], {e, vOrigins[[v]]}], {v, lLists[[l]]}];
   If[l > 1, eRevHashesC[l - 1, m eMaxSize, lLists, vRHashes, eRHashes, vOrigins, eOrigins]]
  ];
SetAttributes[eRevHashesC, HoldAll];
eRevHashesC[l_, m_, lLists_, vRHashes_, eRHashes_, vOrigins_, eOrigins_] := Block[{},
   Do[eRHashes[[e]] += m; Do[vRHashes[[v]] += eRHashes[[e]], {v, eOrigins[[e]]}], {e, lLists[[l]]}];
   If[l > 1, vRevHashesC[l - 1, m vMaxSize, lLists, vRHashes, eRHashes, vOrigins, eOrigins]]
  ];
```

`vertsIsoQ` prüft analog zu `edgesIsoQ` die Isomorphie von zwei Vertices des Hypergraphen,

```
vertsIsoQ[edgeList_, vert1_, vert2_] := edgesIsoQ[dualHG[edgeList], vert1, vert2];
```

Dabei wird ausgenutzt, dass Kanten und Vertices in Hypergraphen äquivalent sind, jedem Hypergraphen ist sein Dualer Hypergraph zugeornet, bei welchem Kanten und Vertices gerade vertauscht sind,

```
dualHG[list_] := Block[{dl},
   dl = Table[{}, {Max[Flatten[list]]}];
   Do[Do[AppendTo[dl[[j]], i], {j, list[[i]]}], {i, Length[list]}];
   Return[dl]];
```

`findIsoEdges` gruppiert die Kanten des Hypergraphen zu Mengen von isomorphen Kanten,

```
findIsoEdges[edgeList_] := Gather[Range[Length[edgeList]], edgesIsoQ[edgeList, #1, #2] &];
```

`findIsoVerts` verfährt entsprechend mit den Vertices,

```
findIsoVerts[edgeList_] := Gather[Range[Length[dualHG[edgeList]]],
   vertsIsoQ[edgeList, #1, #2] &];
```

`hgsIsoQ[hg1EdgeList, hg2EdgeList]` prüft, ob zwei Hypergraphen isomorph sind,

```
hgsIsoQ[hg1EdgeList_, hg2EdgeList_] := Block[
   {vCount, eCount, vList1, vList2, eList1, eList2, e1, e2, eMap, vMap, eOrigins,
    vOrigins, eOrigins2, vOrigins2, levels, levels2, lLists, lLists2, levelsMaps,
    vHashes1, eHashes1, vHashes2, eHashes2, vMaxSize, eMaxSize, eRHashes, vRHashes,
    eRHashes2, vRHashes2, vCHashes, eCHashes, vCHashes2, eCHashes2, eHGroups1,
    eHGroups2, result},

   Catch[
    eList1 = hg1EdgeList; eList2 = hg2EdgeList;
    If[Max[Flatten[eList1]] ≠ Max[Flatten[eList2]] || Length[eList1] ≠ Length[eList2],
     Throw[False]];
    vCount = Max[Flatten[eList1]]; eCount = Length[eList1];
    vList1 = Table[{}, {vCount}]; Do[Do[AppendTo[vList1[[v]], e], {v, eList1[[e]]}],
     {e, eCount}];
    vList2 = Table[{}, {vCount}]; Do[Do[AppendTo[vList2[[v]], e], {v, eList2[[e]]}], {e, eCount}];
    eMap = Table[0, {eCount}]; vMap = Table[0, {vCount}];
    eOrigins = Table[{}, {eCount}]; vOrigins = Table[{}, {vCount}];
    eOrigins2 = Table[{}, {eCount}]; vOrigins2 = Table[{}, {vCount}];
    eRHashes = Table[0, {eCount}]; vRHashes = Table[0, {vCount}];
    eRHashes2 = Table[0, {eCount}]; vRHashes2 = Table[0, {vCount}];
    vMaxSize = Max[Length /@ vList1]; eMaxSize = Max[Length /@ eList1];
    vHashes1 = Table[vHash[v, 2, vList1, eList1], {v, vCount}];
    eHashes1 = Table[eHash[e, 2, vList1, eList1], {e, eCount}];
    vHashes2 = Table[vHash[v, 2, vList2, eList2], {v, vCount}];
    eHashes2 = Table[eHash[e, 2, vList2, eList2], {e, eCount}];
    eHGroups1 = SortBy[GatherBy[Range[eCount], eHashes1[[#1]] &], {Length, eHashes1[[#1[[1]]]] &}];
    eHGroups2 = SortBy[GatherBy[Range[eCount], eHashes2[[#1]] &], {Length, eHashes2[[#1[[1]]]] &}];
    If[Length /@ eHGroups1 ≠ Length /@ eHGroups2, Throw[False]];
    Do[If[eHashes1[[eHGroups1[[g]][[1]]]] ≠ eHashes2[[eHGroups2[[g]][[1]]]], Throw[False]],
     {g, Length[eHGroups1]}];

    e1 = eHGroups1[[1]][[1]]; lLists = {{e1}}; vTreeC[lLists, vOrigins, eOrigins, vList1, eList1];
    If[Mod[Length[lLists], 2] == 0,
     vRevHashesC[Length[lLists], 1, lLists, vRHashes, eRHashes, vOrigins, eOrigins],
     eRevHashesC[Length[lLists], 1, lLists, vRHashes, eRHashes, vOrigins, eOrigins]];
    vCHashes = Table[vHashes1[[v]] + Max[Length /@ vHashes1] vRHashes[[v]], {v, vCount}];
    eCHashes = Table[eHashes1[[e]] + Max[Length /@ eHashes1] eRHashes[[e]], {e, eCount}];
    levels = {}; eGroupsC[];
```

```
  Do[
   eOrigins2 = Table[{}, {eCount}]; vOrigins2 = Table[{}, {vCount}];
   lLists2 = {{e2}}; vTreeC[lLists2, vOrigins2, eOrigins2, vList2, eList2];
   If[Mod[Length[lLists2], 2] == 0,
    vRevHashesC[Length[lLists2], 1, lLists2, vRHashes2, eRHashes2, vOrigins2, eOrigins2],
    eRevHashesC[Length[lLists2], 1, lLists2, vRHashes2, eRHashes2, vOrigins2, eOrigins2]];
   vCHashes2 = Table[vHashes2[[v]] + Max[Length /@ vHashes2] vRHashes2[[v]], {v, vCount}];
   eCHashes2 = Table[eHashes2[[e]] + Max[Length /@ eHashes2] eRHashes2[[e]], {e, eCount}];
   levels2 = Table[{}, {Length[levels]}]; levels2[[1]] = {{e2}};
   levelsMaps = Table[{}, {Length[levels]}];
   Do[levelsMaps[[l]] = Table[{}, {Length[levels[[l]]]}], {l, Length[levels]}];

   eMap[[e2]] = e1; result = checkVLevel[2];
   If[result,
    Do[If[! Sort[Map[vMap[[#1]] &, eList2, {2}][[e]]] == Sort[eList1[[eMap[[e]]]]], Print["Error!"]],
     {e, eCount}];
    Throw[True]],
   {e2, eHGroups2[[1]]}];
  Throw[False]]
];
```

`hgMinorQ[hg1EdgeList, hg2EdgeList]` prüft, ob der durch `hg2EdgeList` gegebene Hypergraph aus dem durch `hg1EdgeList` gegebenen durch hinzufügen von Kanten und Vertices konstruiert werden kann,

```
hgMinorQ[hg1EdgeList_, hg2EdgeList_] := Block[
   {vCount1, eCount1, vCount2, eCount2, vList1, vList2, eList1, eList2, e1, e2, eMap,
    vMap, eMapR, eOrigins1, vOrigins1, eOrigins2, vOrigins2, eSizes1, eSizes2,
    levels1, levels2, lLists1, lLists2, result, nextE2},

   Catch[
    eList1 = hg1EdgeList; eList2 = hg2EdgeList;
    vCount1 = Max[Flatten[eList1]]; eCount1 = Length[eList1];
    vCount2 = Max[Flatten[eList2]]; eCount2 = Length[eList2];
    vList1 = Table[{}, {vCount1}]; Do[Do[AppendTo[vList1[[v]], e], {v, eList1[[e]]}],
     {e, eCount1}];
    vList2 = Table[{}, {vCount2}]; Do[Do[AppendTo[vList2[[v]], e], {v, eList2[[e]]}],
     {e, eCount2}];
    eMap = Table[0, {eCount1}]; vMap = Table[0, {vCount1}];
    eOrigins1 = Table[{}, {eCount1}]; vOrigins1 = Table[{}, {vCount1}];
    eSizes1 = Table[{e, Sort[Table[Length[vList1[[v]]], {v, eList1[[e]]}]]}, {e, eCount1}];
    eSizes2 = Table[{e, Sort[Table[Length[vList2[[v]]], {v, eList2[[e]]}]]}, {e, eCount2}];
    e1 = SortBy[eSizes1, {Length[#1[[2]]] &, #1[[2]][[-1]] &}][[-1]][[1]];
    lLists1 = {{e1}}; vTreeC[lLists1, vOrigins1, eOrigins1, vList1, eList1];
    levels1 = {}; eMGroupsC[];

    Do[
     If[Length[eList1[[e1]]] > Length[eList2[[e2]]], Goto[nextE2]];
     Do[If[eSizes1[[e1]][[2]][[i]] > eSizes2[[e2]][[2]][[i]], Goto[nextE2]], {i, Length[eList1[[e1]]]}];
     eOrigins2 = Table[{}, {eCount2}]; vOrigins2 = Table[{}, {vCount2}];
     lLists2 = {{e2}}; vTreeC[lLists2, vOrigins2, eOrigins2, vList2, eList2];
     eMap[[e1]] = e2; result = checkVMLevel[2];
     If[result, (*printMap[vMap];printMap[eMap];*)Throw[True]];
     Label[nextE2],
     {e2, eCount2}];
    Throw[False]]
  ];
```

Diese Funktion verwendet leicht modifizierte Gruppierungsfunktionen,

```
vMGroupsC[] := Block[{},
   AppendTo[levels1, GatherBy[SortBy[lLists1[[Length[levels1] + 1]],
      Sort[vOrigins1[[#1]] &], Sort[vOrigins1[[#1]] &]];
   If[Length[levels1] < Length[lLists1], eMGroupsC[]]
  ];
eMGroupsC[] := Block[{},
   AppendTo[levels1, GatherBy[SortBy[lLists1[[Length[levels1] + 1]],
      Sort[eOrigins1[[#1]] &], Sort[eOrigins1[[#1]] &]];
   If[Length[levels1] < Length[lLists1], vMGroupsC[]]
  ];
```

,

und veränderte Prüffunktionen,

```
checkVMLevel[l_] :=
  Block[{targetGroups, usedTargets, levelMap, p, nxt, nextGroup, nextMap, try},
   Catch[
    If[l > Length[levels1], Throw[True]];
    targetGroups = Table[{}, {Length[levels1[[l]]}];
    Do[Do[Do[If[! MemberQ[vOrigins2[[v2]], eMap[[e]], Goto[nxt]],
       {e, vOrigins1[[levels1[[l]][[g]][[1]]]}];
      AppendTo[targetGroups[[g]], v2]; Label[nxt],
      {v2, lLists2[[l]]}];
     If[Length[targetGroups[[g]] < Length[levels1[[l]][[g]], Throw[False]],
     {g, Length[targetGroups]}];
    levelMap = Table[Table[i, {i, Length[levels1[[l]][[g]]}], {g, Length[levels1[[l]]}];
    p = Table[1, {Length[levels1[[l]]}];
    While[True,
     Label[try]; usedTargets = {};
     Do[Do[If[MemberQ[usedTargets, targetGroups[[g]][[levelMap[[g]][[i]]]], Goto[nextMap]];
       AppendTo[usedTargets, targetGroups[[g]][[levelMap[[g]][[i]]]],
       {i, Length[levelMap[[g]]}], {g, Length[levelMap]}];
     Do[Do[vMap[[levels1[[l]][[g]][[i]]] = targetGroups[[g]][[levelMap[[g]][[i]]],
       {i, Length[levelMap[[g]]}], {g, Length[levelMap]}];
     If[checkEMLevel[l + 1], Throw[True]];
     Label[nextMap];
     Do[levelMap[[g]] = Permutations[Range[Length[targetGroups[[g]]],
         {Length[levelMap[[g]]}][[p[[g]]]; p[[g]]++;
      If[p[[g]] > Length[Permutations[Range[Length[targetGroups[[g]]],
          {Length[levelMap[[g]]}]], p[[g]] = 1,
       Goto[try]], {g, Length[levelMap]}];
     Throw[False]]]
  ];
```

`vertSetsIsoQ[edgeList, verts1, verts2]` prüft, ob die beiden Mengen `verts1` und

verts2 von Vertices eines durch edgeList gegebenen Hypergraphen isomorph sind,

```
vertSetsIsoQ[edgeList_, verts1_, verts2_] :=
  hgsIsoQ[Append[edgeList, verts1], Append[edgeList, verts2]];
```

edgeSetsIsoQ prüft das entsprechende für Kanten,

```
edgeSetsIsoQ[edgeList_, edges1_, edges2_] := If[Length[edges1] ≠ Length[edges2], False,
   vertSetsIsoQ[dualHG[edgeList], edges1, edges2]];
```

isoEdgeSets[edgeList_, set_] liefert alle Mengen von Kanten, die isomorph zu der gegebenen Kanten-Menge set sind,

```
isoEdgeSets[edgeList_, set_] := First[Last[Reap[
     Do[If[edgeSetsIsoQ[edgeList, set, s], Sow[s]],
      {s, Subsets[Range[Length[edgeList]], Length[set]]}]]]];
```

isoVertSets arbeitet entsprechen für Vertices,

```
isoVertSets[edgeList_, set_] := First[Last[Reap[
     Do[If[vertSetsIsoQ[edgeList, set, s], Sow[s]],
      {s, Subsets[Range[Length[dualHG[edgeList]]], Length[set]]}]]]];
```

3.2.2 Konstruktion der orthogonalen Hypergraphen

consMtis[startMti, vCountMax] konstruiert alle „orthogonalen" Hypergraphen bis zu einer maximalen Zahl vCountMax von Vertices,

```
consMtis[startMti_, vCountMax_] := Block[{},
   finds = {}; steps = {{{addSFs[startMti], {}}}};
   Do[steps[[-1]] = DeleteDuplicates[steps[[-1]], hgsIsoQ[#1[[1]], #2[[1]]] &];
    AppendTo[steps, {}]; Do[steps[[-1]] = Join[steps[[-1]], nextMtis[mti]], {mti, steps[[-2]]}],
    {s, vCountMax - Max[Flatten[startMti]] + 1}];
   Return[finds];
  ];
```

und benutzt

```
nextMtis[mti_] := Block[
   {flats, vects, orthos, fCount, vCount, tmp, vi, vc, fc, nCs, f, gFSets, fs, noFix,
    retry, nMtis},

   nMtis = {}; flats = mti[[1]]; orthos = {};
   fCount = Length[flats]; vCount = Max[Flatten[flats]];
   vects = Table[{}, {vCount}]; Do[Do[AppendTo[vects[[v]], f], {v, flats[[f]]}], {f, fCount}];

   Label[retry];
   vc = fc = 0;
   Do[tmp = 0; Do[vi = (flats[[f]]) ⋂ (flats[[f2]]);
     If[vi ≠ {}, If[Length[flats[[f2]]] > 2 && ! pairListedQ[orthos, v, vi[[1]]], tmp++]],
     {f2, vects[[v]]}];
    If[! MemberQ[flats[[f]], v] && tmp > 0 && Length[flats[[f]]] - tmp > 1, {vc, fc} = {v, f};
     Break[]],
    {f, fCount}, {v, vCount}];

   If[vc == 0,
    AppendTo[finds, {remSFs[flats], orthos}];
    finds = DeleteDuplicates[finds, hgsIsoQ[#1[[1]], #2[[1]]] && #1[[2]] == #2[[2]] &];
    If[vCount == vCountMax, Return[{}]];

    gFSets = Gather[getUnjoinedFlats[], edgeSetsIsoQ[flats, #1, #2] &];
    Do[AppendTo[nMtis, flats]; Do[AppendTo[nMtis[[-1]][[f]], vCount + 1], {f, g[[1]]}];
     nMtis[[-1]] = addSFs[nMtis[[-1]]],
     {g, gFSets}];
    AppendTo[nMtis, flats]; AppendTo[nMtis[[-1]], {1, vCount + 1}];
    nMtis[[-1]] = addSFs[nMtis[[-1]]];
    Return[Table[{nMtis[[i]], orthos}, {i, Length[nMtis]}]],

    If[vCount == vCountMax, Return[{}]];
    Do[vi = (flats[[f]]) ⋂ (flats[[fc]]); If[vi ≠ {}, If[Length[flats[[f]]] == 3, Goto[noFix]];
      If[Length[flats[[f]]] > 3 && ! pairListedQ[orthos, vi[[1]], vc],
       Do[If[pairListedQ[orthos, v2, vi[[1]]] || pairListedQ[orthos, v2, vc], Goto[noFix]],
        {v2, flats[[f]]}];
       AppendTo[orthos, {vi[[1]], vc}]]],
     {f, vects[[vc]]}];
    Goto[retry]; Label[noFix];

    fs = First[Last[Reap[
         Do[If[(flats[[f]]) ⋂ (flats[[fc]]) ≠ {} && Length[flats[[f]]] == 2, Sow[f]],
          {f, vects[[vc]]}]]]];
    gFSets = Gather[Select[getUnjoinedFlats[], #1 ⋂ fs ≠ {} &],
      edgeSetsIsoQ[flats, #1, #2] &];
    Do[AppendTo[nMtis, flats]; Do[AppendTo[nMtis[[-1]][[f]], vCount + 1], {f, g[[1]]}];
     nMtis[[-1]] = addSFs[nMtis[[-1]]],
     {g, gFSets}];
    Return[Table[{nMtis[[i]], orthos}, {i, Length[nMtis]}]];
   ];
  ];
```

und

```
getUnjoinedFlats[] := Block[{usedVs, fSet, sets},
   sets = {{}};
   Do[fSet = {}; usedVs = {}; Do[If[usedVs ⋂ (flats[[f]]) == {},
       AppendTo[fSet, f]; usedVs = usedVs ⋃ flats[[f]]],
     {f, f1, fCount}]; sets = sets ⋃ Subsets[fSet],
    {f1, 1, fCount}];
   Return[Complement[sets, {{}}]];
  ];
```

`consSMtis[sMti, sCount]` konstruiert nur die „orthogonalen" Hypergraphen, welche, von einem gegebenen „orthogonalen" Hypergraphen `sMti` ausgehend, durch das sukzessive Hinzufügen von einzelnen Vektoren in vorhandene, 2 oder mehr Vektoren enthaltende, 2-flats entstehen, unter Beibehaltung der Orthogonalitätseigenschaft bei jedem Schritt,

```
consSMtis[sMti_, sCount_] := Block[{},
   steps = {{{addSFs[sMti], {}}}};
   Do[
    steps[[-1]] = DeleteDuplicates[steps[[-1]], hgsIsoQ[remSFs[#1[[1]]], remSFs[#2[[1]]]] &];
    AppendTo[steps, {}];
    Do[steps[[-1]] = Join[steps[[-1]], nextSMtis[mti]], {mti, steps[[-2]]}],
    {sCount}];
   Return[Map[{remSFs[#1[[1]]], #1[[2]]} &, steps, {2}]];
  ];
```

mit

```
nextSMtis[sMti_] := Block[
   {flats, cFlats, vects, cVects, cOrthos, fCount, vCount, cFCount, cVCount, tmp, vi,
    vc, fc, gFSets, noFix, retry, nMtis},

   nMtis = {}; flats = sMti[[1]]; fCount = Length[flats]; vCount = Max[Flatten[flats]];
   vects = Table[{}, {vCount}]; Do[Do[AppendTo[vects[[v]], f], {v, flats[[f]]}], {f, fCount}];

   gFSets = Gather[getUnjoinedFlats[], edgeSetsIsoQ[flats, #1, #2] &];
   Do[cFlats = flats; Do[AppendTo[cFlats[[f]], vCount + 1], {f, g[[1]]}];
    cFlats = addSFs[cFlats];

    cOrthos = {}; cFCount = Length[cFlats]; cVCount = vCount + 1;
    cVects = Table[{}, {cVCount}];
    Do[Do[AppendTo[cVects[[v]], f], {v, cFlats[[f]]}], {f, cFCount}];
    Label[retry];
    vc = fc = 0;
    Do[tmp = 0; Do[vi = (cFlats[[f]]) ⋂ (cFlats[[f2]]);
      If[vi ≠ {}, If[Length[cFlats[[f2]]] > 2 && ! pairListedQ[cOrthos, v, vi[[1]]], tmp++]],
      {f2, cVects[[v]]}];
     If[! MemberQ[cFlats[[f]], v] && tmp > 0 && Length[cFlats[[f]]] - tmp > 1,
      {vc, fc} = {v, f}; Break[]],
     {f, cFCount}, {v, cVCount}];
    If[vc == 0, AppendTo[nMtis, {cFlats, cOrthos}],

     Do[vi = (cFlats[[f]]) ⋂ (cFlats[[fc]]); If[vi ≠ {}, If[Length[cFlats[[f]]] == 3,
        Goto[noFix]];
       If[Length[cFlats[[f]]] > 3 && ! pairListedQ[cOrthos, vi[[1]], vc],
        Do[If[pairListedQ[cOrthos, v2, vi[[1]]] || pairListedQ[cOrthos, v2, vc],
          Goto[noFix]],
         {v2, cFlats[[f]]}];
        AppendTo[cOrthos, {vi[[1]], vc}]]],
      {f, cVects[[vc]]}];
     Goto[retry]; Label[noFix];],
    {g, gFSets}];
   Return[nMtis]
  ];
```

3.2.3 Alle orthononalen Hypergraphen mit bis zu 10 Vektoren

`consMtis[10]` liefert

1. {{{1,2,3},{1,4,5},{2,4,6},{3,5,6}},{}}

2. {{{1,2,3},{1,4,5},{2,4,6},{2,5,7},{3,4,7},{1,6,7}},{}}
3. {{{1,2,3},{1,4,5},{2,4,6},{2,5,7},{3,4,7},{3,5,6},{1,6,7}},{}}
4. {{{1,2,3},{1,4,5},{2,4,6,7},{2,5,8},{3,4,8},{3,5,7},{1,6,8}},{{7,6}}}
5. {{{1,2,3,9},{1,4,5,8},{2,4,6},{3,5,7},{3,6,8},{5,6,9},{2,7,8},{4,7,9}},{}}
6. {{{1,2,3,8},{1,4,5},{2,4,6,7},{3,5,7,9},{1,6,9},{5,6,8},{8,4,9}},{{2,7},{3,7},{2,3}}}
7. {{{1,2,3,8},{1,4,5},{2,4,6,7},{2,5,9},{3,4,9},{3,5,7},{1,6,9},{5,6,8},{8,7,9}},{}}
8. {{{1,2,3},{1,4,5},{2,4,6},{3,5,6},{7,8,9}},{}}
9. {{{1,2,3,10},{1,4,5,8},{2,4,6,9},{2,5,7},{3,5,9},{3,6,8},{5,6,10},{1,7,9},{4,7,10},{8,9,10}},{{5,8},{9,6},{10,3}}}
10. {{{1,2,3,9},{1,4,5},{2,4,6,10},{2,5,7,8},{3,4,8},{3,5,10},{1,6,8},{5,6,9},{1,7,10},{4,7,9}},{}}
11. {{{1,2,3,9},{1,4,5},{2,4,6,10},{2,5,7,8},{3,4,8},{3,5,10},{1,6,8},{5,6,9},{1,7,10},{4,7,9},{9,8,10}},{}}
12. {{{1,2,3},{1,4,5,8},{2,4,6},{2,5,10},{3,4,10},{3,5,7},{1,6,10},{3,6,8},{5,6,9},{2,7,8},{4,7,9},{8,9,10}},{{4,1}}}
13. {{{1,2,3,9},{1,4,5,8},{2,4,6},{2,5,10},{3,4,10},{3,5,7},{3,6,8},{5,6,9},{2,7,8},{4,7,9},{9,8,10}},{}}
14. {{{1,2,3,9},{1,4,5,8},{2,4,6},{2,5,10},{3,4,10},{3,5,7},{1,6,10},{3,6,8},{5,6,9},{2,7,8},{4,7,9},{9,8,10}},{}}
15. {{{1,2,3,9,10},{1,4,5},{2,4,6},{3,4,8},{3,5,7},{1,6,8},{5,6,9},{2,7,8},{4,7,9},{6,7,10},{5,8,10}},{}}
16. {{{1,2,3,8},{1,4,5,10},{2,4,6,7},{3,5,7,9},{1,6,9},{3,6,10},{5,6,8},{8,4,9},{8,7,10},{2,9,10}},{}}
17. {{{1,2,3,8},{1,4,5},{2,4,6,7},{2,5,9,10},{3,4,9},{3,5,7},{1,6,9},{3,6,10},{5,6,8},{1,7,10},{8,4,10},{8,7,9}},{}}
18. {{{1,2,3},{1,4,5},{2,4,6},{2,5,7},{3,4,7},{1,6,7},{8,9,10}},{}}
19. {{{1,2,3},{1,4,5},{2,4,6},{2,5,7},{3,4,7},{3,5,6},{1,6,7},{8,9,10}},{}}
20. {{{1,2,3},{1,4,5},{2,4,6},{3,5,6},{1,7,8},{2,7,9},{4,7,10},{3,8,9},{5,8,10},{6,9,10}},{}}
21. {{{1,2,3},{1,4,5},{2,4,6},{3,5,6},{7,8,9,10}},{}}

Die erste Menge gibt dabei jeweils die 2-flats mit mehr als 2 Elementen an, die zweite Menge gibt ggf. an, welch Vektoren in 2-flats mit 4 oder mehr Elementen orthogonal sein müssen.

Die Analyse ergibt

1. ist A3
2. sind die Vektor- und Spinorgewichte von B_3
3. ist das Fanomatroid
4. ist (E_6, A_3)
5. ist nicht darstellbar
6. ist B_3
7. ist nicht darstellbar
8. ist reduzibel
9. ist $(E_6, A_1 \times A_2)$
10. ist (E_6, A_1^3)
11. ist nicht darstellbar
12. liefert kein $\vee$-System
13. ist nicht darstellbar
14. ist nicht darstellbar
15. ist nicht darstellbar
16. ist nicht darstellbar
17. ist nicht darstellbar
18. ist reduzibel
19. ist reduzibel
20. ist A_4
21. ist reduzibel

3.3 Realisierung der Hypergraphen

Die Realisierung der Hypergraphen als Vektorsysteme im $\mathbb{R}^n$ erfolgt in zwei Schritten: Im ersten Schritt wird ein System von Einheitsvektoren erzeugt, welches den vom Hypergraphen vorgegebenen Inzidenzen in Ebenen sowie den $\vee$-Bedingungen für die Ebenen mit 2 Vektoren, den Orthogonalitätsbedingungen, entspricht und über die Winkel zwischen ihnen parametrisiert. Im zweiten Schritt wird die Länge der Vektoren durch Lösen der $\vee$-Bedingungen für die Ebenen mit mehr als 2 Vektoren in der Form (2.24) festgelegt. Dabei stellt sich heraus, ob der Hypergraph ein $\vee$-System liefert.

3.3.1 Konstruktion der Einheitsvektoren

die Funktionen müssen noch „von Hand" aufgerufen werden

```
firstV[id_] := Block[{},
   vectors[[id]] = UnitVector[dim, 1]; Do[AppendTo[fStats[[f]], id], {f, vects[[id]]}]
  ];

secondV[id_, ang_] := Block[{v},
   v = ConstantArray[0, dim]; v[[1]] = Cos[ang]; v[[2]] = Sin[ang];
   vectors[[id]] = v; Do[AppendTo[fStats[[f]], id], {f, vects[[id]]}]
  ];

fltAngV[id_, flt_, vec_, ang_] := Block[{cs, s1, s2, sf, n, r, k1, k2, v},
   s1 = vectors[[fStats[[flt]][[1]]].vectors[[vec]]; s2 = vectors[[fStats[[flt]][[2]]].vectors[[vec]];
   sf = vectors[[fStats[[flt]][[1]]].vectors[[fStats[[flt]][[2]]]; cs = Cos[ang];
   (*FullSimplify[Solve[{k1 s1+k2 s2==cs,k1^2+k2^2+2 k1 k2 sf==1},{k1,k2}]]*)
   n = s1^2 + s2^2 - 2 s1 s2 sf; r = Sqrt[n - cs^2 + cs^2 sf^2];
   k1 = (cs (s1 - s2 sf) - sign[ang] sign[s1] s2 r) / n;
   k2 = (cs (s2 - s1 sf) + sign[ang] Abs[s1] r) / n;
   v = k1 vectors[[fStats[[flt]][[1]]] + k2 vectors[[fStats[[flt]][[2]]];
   vectors[[id]] = Simplify[v]; Do[AppendTo[fStats[[f]], id], {f, vects[[id]]}]
  ];

otgsAngV[id_, oVecs_, vec_, ang_] := Block[{pb1, pb2, cs, s1, s2, n, r, k1, k2, v},
   pb1 = Normalize[Cross @@ Append[(vectors[[#1]] &) /@ oVecs, vectors[[vec]]]];
   pb2 = Normalize[Cross @@ Append[(vectors[[#1]] &) /@ oVecs, pb1]];
   s1 = pb1.vectors[[vec]]; s2 = pb2.vectors[[vec]]; cs = Cos[ang];
   n = s1^2 + s2^2; r = Sqrt[n - cs^2];
   k1 = (cs s1 - sign[ang] sign[s1] s2 r) / n; k2 = (cs s2 + sign[ang] Abs[s1] r) / n;
   v = k1 pb1 + k2 pb2;
   vectors[[id]] = Simplify[v]; Do[AppendTo[fStats[[f]], id], {f, vects[[id]]}]
  ];
```

```
fltFltV[id_, flt1_, flt2_] := Block[{kv, v},
   kv =
    LinearSolve[{vectors[[fStats[[flt1]][[1]]], vectors[[fStats[[flt1]][[2]]],
       vectors[[fStats[[flt2]][[1]]]}^T,
     vectors[[fStats[[flt2]][[2]]]];
   v = Normalize[kv[[1]] * vectors[[fStats[[flt1]][[1]]] + kv[[2]] * vectors[[fStats[[flt1]][[2]]]];
   vectors[[id]] = Simplify[v]; Do[AppendTo[fStats[[f]], id], {f, vects[[id]]}];
  ];
fltOtgV[id_, flt_, oVec_] := Block[{s1, s2, sf, n, v},
   s1 = vectors[[fStats[[flt]][[1]]].vectors[[oVec]]; s2 = vectors[[fStats[[flt]][[2]]].vectors[[oVec]];
   sf = vectors[[fStats[[flt]][[1]]].vectors[[fStats[[flt]][[2]]];
   n = Sqrt[s1^2 + s2^2 - 2 * s1 * s2 * sf];
   v = -vectors[[fStats[[flt]][[1]]] * s2 / n + vectors[[fStats[[flt]][[2]]] * s1 / n;
   vectors[[id]] = Simplify[v]; Do[AppendTo[fStats[[f]], id], {f, vects[[id]]}];
  ];
angsV[id_, vecs_, angs_] := Block[{v1, v2, vo, v, ca1, ca2, xy, a, b, g},
   vo = Cross @@ (vectors[[#]] &) /@ vecs;
   (*FullSimplify[Solve[{a+b xy==ca1,a xy+b ==ca2,a^2+b^2+g^2+2 a b xy==1},{a,b,g}]]*)
   ca1 = Cos[angs[[1]]]; ca2 = Cos[angs[[2]]]; xy = vectors[[vecs[[1]]].vectors[[vecs[[2]]];
   a = (-ca1 + ca2 * xy) / (-1 + xy^2); b = (-ca2 + ca1 * xy) / (-1 + xy^2);
   g = Sqrt[(-1 + ca1^2 + ca2^2 - 2 * ca1 * ca2 * xy + xy^2) / (-1 + xy^2)];
   v = a * vectors[[vecs[[1]]] + b * vectors[[vecs[[2]]] + g * vo;
   vectors[[id]] = Simplify[v]; Do[AppendTo[fStats[[f]], id], {f, vects[[id]]}];
  ];
```

3.3.2 Bestimmung der Längen

Die Längen der Kovektoren werden nun aus den $\vee$-Bedingungen bestimmt

```
solveLV[] := Block[{finishedVs, remainingFs, rawVs, fittedVs, rcrs, dc, ds, ar1, ar2, r1, r2},
   multips = Table[1, {Length[vectors]}];
   finishedVs = {}; remainingFs = Range[Length[flats]];
   Do[If[Length[f] == 3, AppendTo[finishedVs, f[[1]]]; Break[]], {f, flats}];
   While[remainingFs ≠ {}, Do[rawVs = Select[flats[[f]], ! MemberQ[finishedVs, #] &];
     If[1 ≤ Length[rawVs] ≤ 2,
      fittedVs = Select[flats[[f]], MemberQ[finishedVs, #] &];
      If[Length[rawVs] == 1,
       AppendTo[rawVs, fittedVs[[-1]]]; fittedVs = Delete[fittedVs, -1];
       finishedVs = DeleteCases[finishedVs, rawVs[[2]]]];
      rcrs = Cross[vectors[[flats[[f]][[1]]]], vectors[[flats[[f]][[2]]]]];
      dc = Simplify[Sum[multips[[v]] *
           Cos[2 * Sign[rcrs.Cross[vectors[[v]], vectors[[fittedVs[[1]]]]] *
             VectorAngle[vectors[[v]], vectors[[fittedVs[[1]]]]], {v, fittedVs}]];
      ds = Simplify[Sum[multips[[v]] *
           Sin[2 * Sign[rcrs.Cross[vectors[[v]], vectors[[fittedVs[[1]]]]] *
             VectorAngle[vectors[[v]], vectors[[fittedVs[[1]]]]], {v, fittedVs}]];
      ar1 = Simplify[Sign[rcrs.Cross[vectors[[rawVs[[1]]]], vectors[[fittedVs[[1]]]]]
         * VectorAngle[vectors[[rawVs[[1]]]], vectors[[fittedVs[[1]]]]];
      ar2 = Simplify[Sign[rcrs.Cross[vectors[[rawVs[[2]]]], vectors[[fittedVs[[1]]]]]
         * VectorAngle[vectors[[rawVs[[2]]]], vectors[[fittedVs[[1]]]]];
      If[Abs[ar1 - ar2] === Pi / 2,
       multips[[rawVs[[1]]]] = 1; multips[[rawVs[[2]]]] = 1; Print["pi/2!"],
       multips[[rawVs[[1]]]] = Simplify[Csc[2 ar1 - 2 ar2] * (-ds Cos[2 ar2] + dc Sin[2 ar2])];
       multips[[rawVs[[2]]]] = Simplify[Csc[2 ar1 - 2 ar2] * (ds Cos[2 ar1] - dc Sin[2 ar1])]];
      AppendTo[finishedVs, rawVs[[1]]]; AppendTo[finishedVs, rawVs[[2]]];
      remainingFs = DeleteCases[remainingFs, f];
      Continue[]];
     If[Length[rawVs] == 0, remainingFs = DeleteCases[remainingFs, f]],
     {f, remainingFs}]]
  ];
```

3.3.3 Beispiel für einen realisierbaren „orthogonalen" Hypergraphen, der kein ∨-System liefert

Hierbei wird gleichzeitig die Verwendung der Konstruktions-Funktionen demonstriert,

```
flats = u12;
fCount = Length[flats]; vCount = Max[Flatten[flats]];
vects = Table[{}, {vCount}]; Do[Do[AppendTo[vects[[v]], f], {v, flats[[f]]}], {f, fCount}];
dim = 3;
vectors = ConstantArray[0, {vCount, dim}];
fStats = Table[{}, {fCount}]; vStats = Table[{}, {vCount}];
$Assumptions = 0 < a < Pi / 2 && 0 < b < Pi / 2;

firstV[4]

secondV[1, Pi / 2]

otgsAngV[7, {1}, 4, 2 ArcTan[Sqrt[3] - Sqrt[2]]]

fltAngV[9, 11, 4, -2 ArcTan[Sqrt[3] - Sqrt[2]]]

otgsAngV[6, {7}, 1, b]

otgsAngV[10, {7}, 1, -b]

otgsAngV[2, {9}, 1, b]

otgsAngV[3, {9}, 1, -b]

fltFltV[5, 2, 4]

fltFltV[8, 2, 10]

vectors = FullSimplify[vectors]
```

$$\left\{\{0, 1, 0\}, \left\{-\frac{\text{Sin}[b]}{\sqrt{3}}, \text{Cos}[b], \sqrt{\frac{2}{3}}\ \text{Sin}[b]\right\},\right.$$

$$\left\{\frac{\text{Sin}[b]}{\sqrt{3}}, \text{Cos}[b], -\sqrt{\frac{2}{3}}\ \text{Sin}[b]\right\}, \{1, 0, 0\}, \left\{-\frac{\text{Sin}[b]}{\sqrt{2 + \text{Cos}[2\,b]}}, \frac{\sqrt{3}\ \text{Cos}[b]}{\sqrt{2 + \text{Cos}[2\,b]}}, 0\right\},$$

$$\left\{\frac{\text{Sin}[b]}{\sqrt{3}}, \text{Cos}[b], \sqrt{\frac{2}{3}}\ \text{Sin}[b]\right\}, \left\{\sqrt{\frac{2}{3}}, 0, \text{Sin}\left[2\,\text{ArcTan}\left[\sqrt{2} - \sqrt{3}\right]\right]\right\},$$

$$\left.\left\{\frac{\text{Sin}[b]}{\sqrt{2 + \text{Cos}[2\,b]}}, \frac{\sqrt{3}\ \text{Cos}[b]}{\sqrt{2 + \text{Cos}[2\,b]}}, 0\right\}, \left\{\sqrt{\frac{2}{3}}, 0, \frac{1}{\sqrt{3}}\right\}, \left\{-\frac{\text{Sin}[b]}{\sqrt{3}}, \text{Cos}[b], -\sqrt{\frac{2}{3}}\ \text{Sin}[b]\right\}\right\}$$

```
vectors[[7, 3]] = -1/Sqrt[3];
```

dabei wurde

```
Reduce[{-Sin[a] Sin[b]/Sqrt[Abs[Sin[a] Sin[b]]^2 + Cos[b]^2], Cos[b]/Sqrt[Abs[Sin[a] Sin[b]]^2 + Cos[b]^2], 0} ==
 {-Cos[2 a] Csc[a] Sin[b]/Sqrt[Cos[b]^2 + Cos[2 a]^2 Csc[a]^2 Sin[b]^2], Cos[b]/Sqrt[Cos[b]^2 + Cos[2 a]^2 Csc[a]^2 Sin[b]^2], 0},
 {a, b}, Reals]
```

```
FullSimplify[2 ArcTan[Root[1 - 10 #1^2 + #1^4 &, 2]]]
```

$$2\,\text{ArcTan}\left[\sqrt{2}-\sqrt{3}\right]$$

```
N[2 ArcTan[Root[1 - 10 #1^2 + #1^4 &, 2]]]
```

```
-0.61548
```

benutzt

3.4 Klassifikation über partiell geordnete Mengen

partielle Ordnung einer Menge:

```
pOrder[list_, relation_] :=
  Block[{set = list, rel = relation, lMinors, sExtens, roots, leaves, higher, lower},
   lMinors = Table[{}, {Length[set]}]; sExtens = Table[{}, {Length[set]}];
   roots = {}; leaves = {};
   Do[higher = False; lower = False; (*Print[i,roots,leaves];Print[lMinors,sExtens];*)
    Do[If[tryExtens[i, r], higher = True], {r, roots}]; If[! higher, AppendTo[roots, i]];
    Do[If[tryMinor[i, l], lower = True], {l, leaves}]; If[! lower, AppendTo[leaves, i]],
    {i, Length[set]}];
   Return[sExtens];
  ];
```

mit

```
tryExtens[i_, m_] := Block[{higher},
   If[! rel[set[[m]], set[[i]]], Return[False],
    higher = False; Do[If[! i == e, If[tryExtens[i, e], higher = True]], {e, sExtens[[m]]}];
    If[! higher,
     Do[If[rel[set[[i]], set[[e]]], sExtens[[m]] = DeleteCases[sExtens[[m]], e];
       lMinors[[e]] = DeleteCases[lMinors[[e]], m]],
      {e, sExtens[[m]]}];
     If[sExtens[[m]] == {}, leaves = DeleteCases[leaves, m]];
     If[! i == m, AppendTo[sExtens[[m]], i]; AppendTo[lMinors[[i]], m]]];
    Return[True]]
  ];
```

und

```
tryMinor[i_, e_] := Block[{lower},
   If[! rel[set[[i]], set[[e]]], Return[False],
    lower = False; Do[If[! i == m, If[tryMinor[i, m], lower = True]], {m, lMinors[[e]]}];
    If[! lower,
     Do[If[rel[set[[m]], set[[i]]], sExtens[[m]] = DeleteCases[sExtens[[m]], e];
       lMinors[[e]] = DeleteCases[lMinors[[e]], m]],
      {m, lMinors[[e]]}];
     If[lMinors[[e]] == {}, roots = DeleteCases[roots, e]];
     If[! i == e, AppendTo[lMinors[[e]], i]; AppendTo[sExtens[[i]], e]]];
    Return[True]]
  ];
```

Z.B. ergibt

```
pOrder[simple3DHGs, hgMinorQ[#1, #2] &]
```

```
{{2}, {9, 10}, {7}, {}, {}, {4, 5}, {6}, {6}, {3, 8}, {}}
```

mit simple3DRs := {A3, vsB3, B3, F3, G3, AB4A11, AB4A12, E6A13, E6A3, H3};

Ausblick

Als Ziel meiner Diplomarbeit hat sich die Suche nach WDVV-Kovektor-Lösungen bzw. $\vee$-Systemen und der Versuch ihrer Klassifikation ergeben. Der aussichstreichste Weg zum Finden weiterer Lösungen, ja der einzige mir bekannte, mit dem die Hoffnung auf eine vollständige Klassifikation besteht, scheint der Zugang über die Matroide der Systeme zu sein.

Im Hinblick auf die begrenzte mir zur Verfügung stehende Zeit habe ich mich bei der Anfertigung dieser Diplomarbeit dazu entschieden, den Überblick über die physikalische Bedeutung der WDVV-Gleichungen in Kapitel 1 zu einer gewissen Vollständigkeit zu bringen und die Zusammenstellung wichtiger Beweise zu den bisherigen Methoden zur Auffindung von $\vee$-Systemen, sowie insbesondere die vollständige Darstellung des Beweises zu der für die Gewinnung von WDVV-Lösungen aus der Matroid/Hypergraphen-Methode entscheidenden Äquivalenz der WDVV-Gleichungen zu Veselovs $\vee$-Bedingungen in Kapitel 2, fertigzustellen. Ich denke, dass die Kapitel 1 und 2 so möglicherweise für jemand, der sich den WDVV-Gleichungen beschäftigen und die Suche nach Kovektor-Lösungen fortführen möchte, als Motivation und Ausgangspunkt nützlich sind.

Im Gegenzug konnte ich die entwickelte Hypergraphen-Methode nur sehr knapp beschreiben, ich denke aber dass es dennoch zur Not möglich ist die Algorithmen so, wie sie hier abgedruckt und als funktionsfähiges Mathematica-"Notebook" auf der beiliegenden CD enthalten sind, zu verstehen. Sinnvoller wäre natürlich die persönliche Erläuterung der zugrunde liegenden Ideen.

Literatur

[1] E. Witten, *On the structure of the phase of two-dimensional gravity,* Nucl. Phys. B340 (1990) 281

[2] R. Dijkgraaf, H. Verlinde, E. Verlinde, *Topological Strings in $d < 1$*, Nucl. Phys. B352 (1991) 59

[3] B. Dubrovin, *Geometry of 2D topological field theories*, Nucl. Phys. B379 (1992) 627 [hep-th/9407018]

[4] I. Krichever, *The τ-function of the universal Whitham hierarchy, matrix models and topological field theories,* [hep-th/9205110]

[5] M. Kontsevich, Yu. Manin, *Gromov-Witten classes, quantum cohomology, and enumerative geometry,* [hep-th/9402147]

[6] B. Totaro, *The curvature of a Hessian metric,* [math/0401381]

[7] H. Shima, K Yagi, *Geometry of Hessian manifolds*, Differential Geometry and its Applications 7 (1997) 277-290

[8] H. Kito, *On Hessian structures on the Euclidean space and the hyperbolic space,* Osaka J. Math 36 (1) (1999), 51-62.

[9] J. van de Leur, *Twisted GL_n Loop Group Orbit and Solutions of the WDVV Equations*, [nlin/0004021]

[10] H. Aratyn, J. van de Leur, *Solutions of the WDVV Equations and Integrable Hierarchies of KP type*, [hep-th/0104092]

[11] H. Aratyn, J. van de Leur, *Integrable Structure behind WDVV Equations*, [hep-th/0111243]

[12] H. Aratyn, J.F. Gomes, J.W. van de Leur, A.H. Zimerman, *WDVV Equations, Darboux-Egoroff Metric and the Dressing Method,* [math-ph/0210038]

[13] H. Aratyn, J. van de Leur, *The CKP hierarchy and the WDVV prepotential,* [nlin/0302004]

[14] G. Bonelli, M. Matone, *Nonperturbative relations in n=2 SUSY Yang-Mills and WDVV equation,* [hep-th/9605090]

[15] A. Marshakov, A. Mironov, A. Morozov, *WDVV-like equations in $N = 2$ SUSY Yang-Mills Theory,* [hep-th/9607109]

[16] A. Marshakov, A. Mironov, A. Morozov, *WDVV Equations from Algebra of Forms,* [hep-th/9701014]

[17] A. Marshakov, A. Mironov, A. Morozov, *More Evidence for the WDVV Equations in N=2 SUSY Yang-Mills Theories,* [hep-th/9701123]

[18] A. Mironov, *WDVV Equations in Seiberg-Witten theory and associative algebras,* [hep-th/9704205]

[19] J. M. Isidro, *On the WDVV Equation and M-Theory*, [hep-th/9805051]

[20] R. Martini, P.K.H. Gragert, *Solutions of WDVV equations in Seiberg-Witten theory from root systems,* [hep-th/9901166]

[21] S. Bellucci, A.V. Galajinsky, E. Latini, *New insight into WDVV equation*, [hep-th/0411232]

[22] N. Wyllard, *(Super)conformal many-body quantum mechanics with extended supersymmetry*, [hep-th/9910160]

[23] G.W. Gibbons, P.K. Townsend, *Black Holes and Calogero Models*, [hep-th/9812034]

[24] A. Galajinsky, O. Lechtenfeld, K. Polovnikov, *N=4 superconformal Calogero models*, arXiv:0708.1075 [hep-th]

[25] A. Galajinsky, O. Lechtenfeld, K. Polovnikov *N=4 mechanics, WDVV equations and roots*, arXiv:0802.4386 [hep-th]

[26] O. Lechtenfeld, *WDVV solutions from orthocentric polytopes and Veselov systems*, arXiv:0805.3245 [hep-th]

[27] O. Lechtenfeld, *N=4 Mechanics, WDVV Equations and Polytopes*, arXiv:0811.0021 [hep-th]

[28] O. Lechtenfeld, K. Polovnikov, *A new class of solutions to the WDVV equation*, arXiv:0907.2244 [hep-th]

[29] A. P. Veselov, *Deformations of the root systems and new solutions to generalised WDVV equations*, [hep-th/9902142]

[30] A. P. Veselov, *On geometry of a special class of solutions to generalised WDVV equations*, [hep-th/0105020]

[31] A. P. Veselov, *On generalizations of the Calogero–Moser–Sutherland quantum problem and WDVV equations*, [math-ph/0204050]

[32] O. A. Chalykh, A. P. Veselov, *Locus configurations and $\vee$-systems,* [math-ph/0105003]

[33] M. V. Feigin, A. P. Veselov, *Logarithmic Frobenius structures and Coxeter discriminants*, [math-ph/0512095]

[34] M. V. Feigin, A. P. Veselov, *On the geometry of $\vee$-systems,* arXiv:0710.5729 [math-ph]

[35] J. E. Humphreys, *Reflection groups and Coxeter groups*, Cambridge University Press (1990)

[36] V. Serganova, *On generalizations of root systems,* Commun. in Algebra 24 (13) (1996) 4281–4299

[37] A. N. Sergeev, A. P. Veselov, *Deformed quantum Calogero-Moser systems and Lie superalgebras,* [math-ph/0303025]

[38] Johannes Thürigen, *Superkonforme Vielteilchenmechanik und die WDVV-Gleichungen,* [http://www.itp.uni-hannover.de/~lechtenf/theses.html] (2010)

[39] J. G. Oxley, *Matroid Theory,* Oxford University Press (1992)

[40] J. G. Oxley, *What is a Matroid*, [http://www.math.lsu.edu/~oxley/] (2007)

[41] [http://www.wolfram.com/products/mathematica/index.html]

[42] [http://axiom-developer.org/], siehe dort insbesondere das Buch
R. D. Jenks, R. Sutor, *Axiom: The Scientific Computation System*

[43] [http://maxima.sourceforge.net/]

Danksagung

Ich möchte mich bei Herrn Prof. Olaf Lechtenfeld für die Möglichkeit bedanken, diese Arbeit unter seiner Betreuung anfertigen zu können und insbesondere für die Freiheit, die er mir dabei gelassen hat. Weiterhin gilt mein Dank Johannes Thürigen für die gute Zusammenarbeit.

Bei meinen Eltern Gerhard und Cordelia Schwerdtfeger möchte ich mich für die Unterstützung während meines gesamten Studiums und davor bedanken.